Software Defined Networking(SDN)

Introduction

- SDN stands for Software Defined Network which is a networking architecture approach.
- It enables the control and management of the network using software applications.
- SDN is the decoupling of the network control logic from the devices performing the function, such as routers, which control the movement of information in the underlying network.
- Simplifies the management of infrastructure, which may be specific to one organization or partitioned to be shared among several.

Traditional Networking

- Networking has always been very traditional.

- We have specific network devices like routers, switches, and firewalls that are used for specific tasks.

- These network devices are sold by networking vendors like Cisco and often use proprietary hardware.

- Most of these devices are primarily configured through the CLI, although there are some GUI products like CCP (Cisco Configuration Protocol) for the routers or ASDM (Adaptive Security Device Manager) for the Cisco ASA (Adaptive Security Appliance) firewalls.

- A network device, for example, a router has different functions that it has to perform. Think for a moment about some of the things that a router has to do in order to forward an IP packet:
  - It has to check the destination IP address in the routing table in order to figure out where to forward the IP packet to.
  - Routing protocols like OSPF, EIGRP or BGP are required to learn networks that are installed in the routing table.
  - It has to use ARP to figure out the destination MAC address of the next hop or destination and change the destination MAC address in the Ethernet frame.
  - The TTL (Time to Live) in the IP packet has to be decreased by 1 and the IP header checksum has to be recalculated.
  - The Ethernet frame checksum has to be recalculated.

All these different tasks are separated by different **planes**. There are three planes:

1. **Control Plane**

- The control plane is responsible for exchanging routing information, building the ARP table, etc. Here are some tasks that are performed by the control plane:
- Learning MAC addresses to build a switch MAC address table.
- Running STP to create a loop-free topology.
- Building ARP tables.
- Running routing protocols like OSPF, EIGRP, and BGP and building the routing table.

2. Data Plane

The data plane is responsible for forwarding traffic. It relies on the information that the control plane supplies. Here are some tasks that the data plane takes care of:

- Encapsulate and de-encapsulate packets.
- Adding or removing headers like the 802.1Q header.
- Matching MAC addresses for forwarding.
- Matching IP destinations in the routing table.
- Change source and destination addresses when using NAT.
- Dropping traffic because of access-lists.

3. Management Plane

- The management plane is used for access and management of our network devices. For example, accessing our device through telnet, SSH or the console port.

- When discussing SDN, the control and data plane are the most important.

- The best routes are installed in the routing table. Another table that the router has to build is the ARP table.

- Information from the routing and ARP table is then used to build the forwarding table. When the router receives an IP packet, it will be able to forward it quickly since the forwarding table has already been built.

Why SDN is Important?

- **Better Network Connectivity:** SDN provides very better network connectivity for sales, services, and internal communications. SDN also helps in faster data sharing.
- **Better Deployment of Applications:** Deployment of new applications, services, and many business models can be speed up using Software Defined Networking.
- **Better Security:** Software-defined network provides better visibility throughout the network. Operators can create separate zones for devices that require different levels of security. SDN networks give more freedom to operators.
- **Better Control with High Speed:** Software-defined networking provides better speed than other networking types by applying an open standard software-based controller.

Components of Software Defining Networking (SDN)

The three main components that make the SDN are:

1. **SDN Applications:** SDN Applications relay requests or networks through SDN Controller using API.

2. **SDN controller:** SDN_Controller collects network information from hardware and sends this information to applications.

3. **SDN networking devices:** SDN Network devices help in forwarding and data processing tasks.

Fundamental characteristics of SDN

- **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.

SDN Building blocks/ SDN Architecture

Key Architectural Components of Software Defined Networking

1. Application

- The application component consists of programs that communicate with the controller using APIs.
- This component transmits data about desired network behavior and required resources to the controller, creating an overview of the network status in the process.
- The application layer also collects data from the controller layer to make the required decisions for fulfilling application goals.
- Examples of applications include analytics, networking management, and business processes for data center operations. For instance, an analytics application can be configured to bolster network security by recognizing suspicious activity.

2. Controller

- The controller component receives requirements and instructions from the application component and uses logic to process and relay them to the networking layer.

- This core element of the SDN architecture enables centralized supervision and management, enforcement of network policies, and automation across both virtual and physical network environments.

- The controller is also responsible for collecting data about network health and status from the hardware layer and communicating this information to the application component.

- This allows the application component to create an abstract network overview that includes statistics and events.

3. Datapath

- The datapath component allows users to supervise and exert control over the forwarding and processing of information by the hardware layer.
- This layer consists of a control-data-plane interface (CDPI) agent and a traffic-forwarding module and may also contain modules for network traffic processing.
- A single network device can contain one or more SDN datapaths. Likewise, a single SDN datapath may be defined across multiple devices.
- This component can also help with processes such as management of shared hardware, logical to physical mapping, data path slicing or virtualization, and compatibility with non-SDN networking.

4. Control to data-plane interface

- The CDPI is used as an interface between the controller component and the datapath component.

- Its functions include allowing forwarding operations to be programmed, reporting network statistics, and notifying users of events of interest.

- Leading SDN solutions feature CDPI components that are open, interoperable, and vendor-neutral.

5. Northbound interface (NBI)

- The NBI relays data between the controller component, the application component, and the policy layer.

- This component typically provides an abstract view of the network and enables the direct expression of network requirements and behavior, regardless of latitude (abstraction) and longitude (functionality).

6. Southbound interface (SBI)

- The SBI relays data between the controller component and individual hardware units connected to the network, such as routers, access points, switches, and hardware firewalls.

- This component further classifies network concepts into more granular technical details meant for the lower layer of the architecture.

- Simply put, SBIs enable network components to exchange data with lower-level components such as physical and virtual switches, routers, and nodes. For instance, routers rely on the SBI to view the network topology, decide network flow, and execute requests received from the NBI.

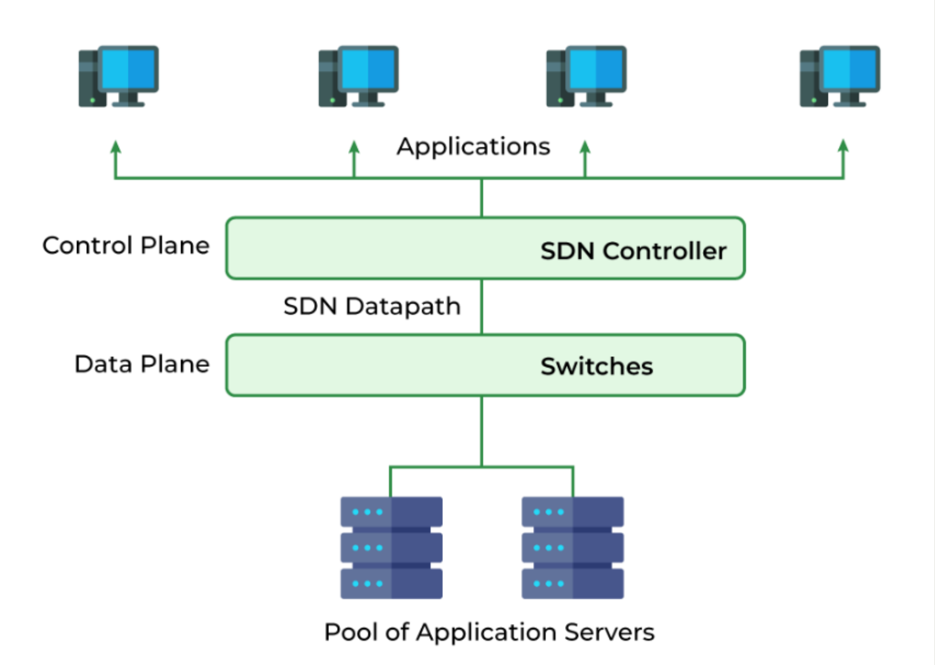
A typical SDN architecture consists of three layers.

•**Application layer:** It contains the typical network applications like intrusion detection, firewall, and load balancing

•**Control layer:** It consists of the SDN controller which acts as the brain of the network. It also allows hardware abstraction to the applications written on top of it.

•**Infrastructure layer:** This consists of physical switches which form the data plane and carries out the actual movement of data packets.

Architecture



SDN Operation

- In a traditional network, each switch has its own data plane as well as the control plane.
- The control plane of various switches exchange topology information and hence construct a forwarding table that decides where an incoming data packet has to be forwarded via the data plane.
- Software-defined networking (SDN) is an approach via which we take the control plane away from the switch and assign it to a centralized unit called the SDN controller.
- Hence, a network administrator can shape traffic via a centralized console without having to touch the individual switches.
- The data plane still resides in the switch and when a packet enters a switch, its forwarding activity is decided based on the entries of flow tables, which are pre-assigned by the controller.

- Flow table consists of match fields (like input port number and packet header) and instructions.

- The packet is first matched against the match fields of the flow table entries.

- Then the instructions of the corresponding flow entry are executed.

- The instructions can be forwarding the packet via one or multiple ports, dropping the packet, or adding headers to the packet.

- If a packet doesn’t find a corresponding match in the flow table, the switch queries the controller which sends a new flow entry to the switch.

- The switch forwards or drops the packet based on this flow entry

Difference between SDN and Traditional Networking

Software Defined Networking	Traditional Networking
Software Defined Network is a virtual networking approach.	A traditional network is the old conventional networking approach.
Software Defined Network is centralized control.	Traditional Network is distributed control.
This network is programmable.	This network is nonprogrammable.
Software Defined Network is the open interface.	A traditional network is a closed interface.
In Software Defined Network data plane and control, the plane is decoupled by software.	In a traditional network data plane and control plane are mounted on the same plane.

Advantages of SDN

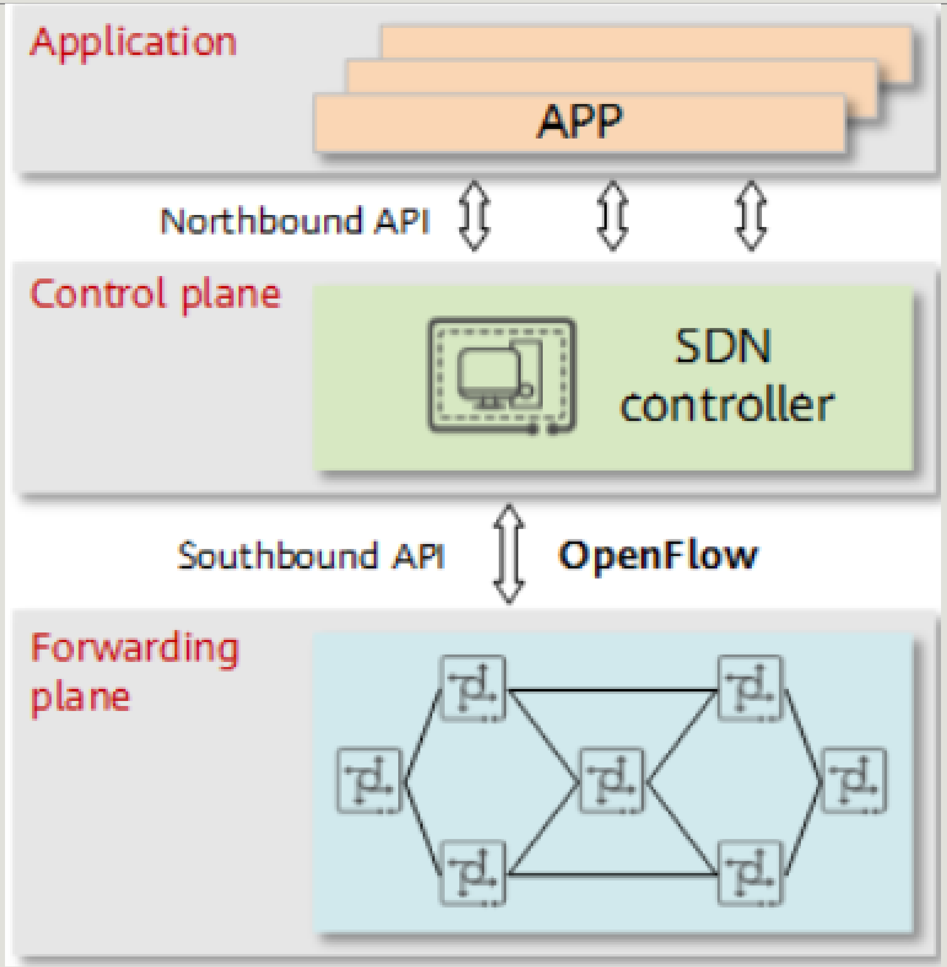
- The network is programmable and hence can easily be modified via the controller rather than individual switches.
- Switch hardware becomes cheaper since each switch only needs a data plane.
- Hardware is abstracted, hence applications can be written on top of the controller independent of the switch vendor.
- Provides better security since the controller can monitor traffic and deploy security policies. For example, if the controller detects suspicious activity in network traffic, it can reroute or drop the packets.
- **Disadvantages of SDN**
- The central dependency of the network means a single point of failure, i.e. if the controller gets corrupted, the entire network will be affected.
- The use of SDN on large scale is not properly defined and explored.

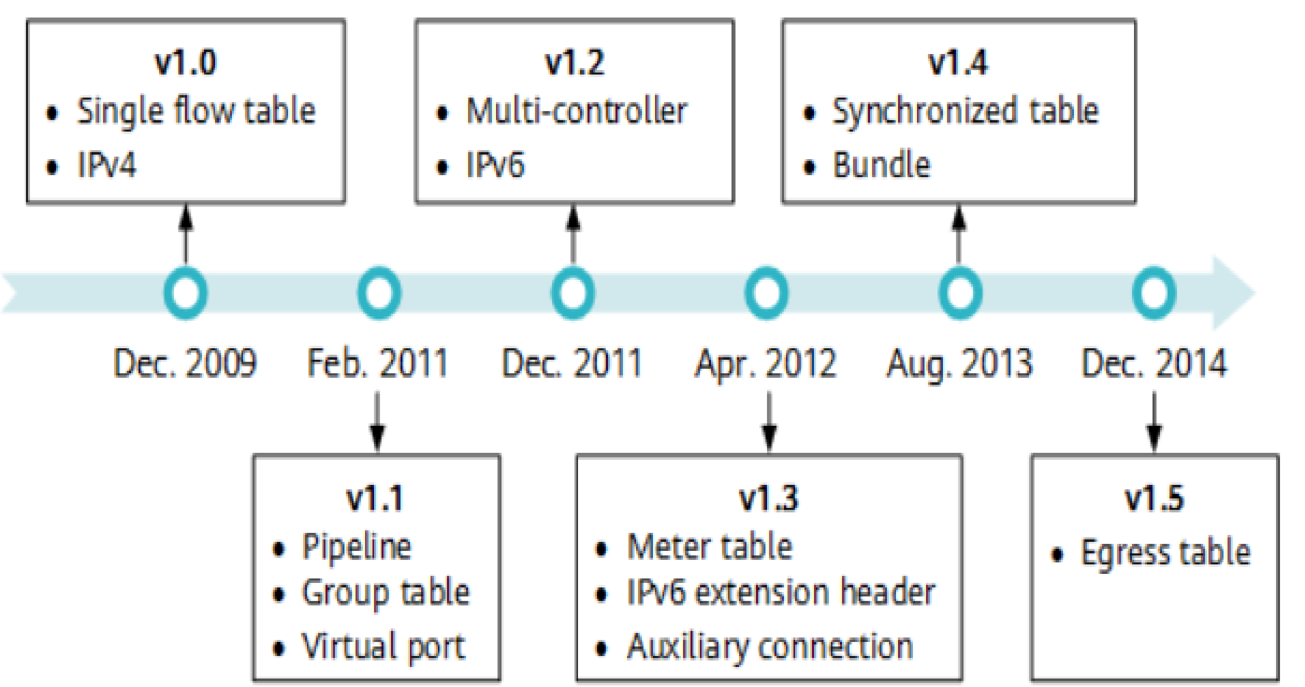
CONTROL PLANE	DATA PLANE
Control plane refers to the all functions and processes that determine which path to use to send the packet or frame.	Data plane refers to all the functions and processes that forward packets/frames from one interface to another based on control plane logic.
It is responsible for building and maintaining the IP routing table.	It is responsible for forwarding actual IP packet.
Control plane responsible about how packets should be forwarded.	Data plane responsible for moving packets from source to destination.
Control plane performs its task independently.	Data plane performs its task depending on Control plane.
In general we can say in control plane it is learned what and how it can be done.	In general we can say in data plane the actual task is performed based on what is learned.
Control plane packets are processed by router to update the routing table.	The forwarding plane/data plane forwards the packets based on the built logic of control plane.
It includes Spanning Tree Protocol (STP), Address Resolution Protocol (ARP), Routing Information Protocol (RIP), Dynamic Host Configuration Protocol (DHCP) etc.	It includes decrementing Time To Live (TTL), recomputing IP header checksum etc.
Control plane packets are locally originated by the router itself.	Data plane packets go through the router.
Control plane acts as a decision maker in data forwarding.	Data plane acts as a decision implementer in data forwarding.
Routing is performed in the control plane.	Switching is performed in the data plane.

What Is OpenFlow?

- OpenFlow is a network communication protocol used between controllers and forwarders in an SDN architecture.
- The core idea of SDN is to separate the forwarding plane from the control plane.
- To achieve this, a communication standard must be built between controllers and forwarders to allow the controllers to directly access and control the forwarding plane of forwarders.
- OpenFlow introduces the concept of flow table, based on which forwarders forwards data packets. Controllers deploy flow tables on forwarders through OpenFlow interfaces, achieving control on the forwarding plane.

OpenFlow in the SDN architecture





Dec. 2009

v1.0
- Single flow table
- IPv4

Feb. 2011

v1.1
- Pipeline
- Group table
- Virtual port

Dec. 2011

v1.2
- Multi-controller
- IPv6

Apr. 2012

v1.3
- Meter table
- IPv6 extension header
- Auxiliary connection

Aug. 2013

v1.4
- Synchronized table
- Bundle

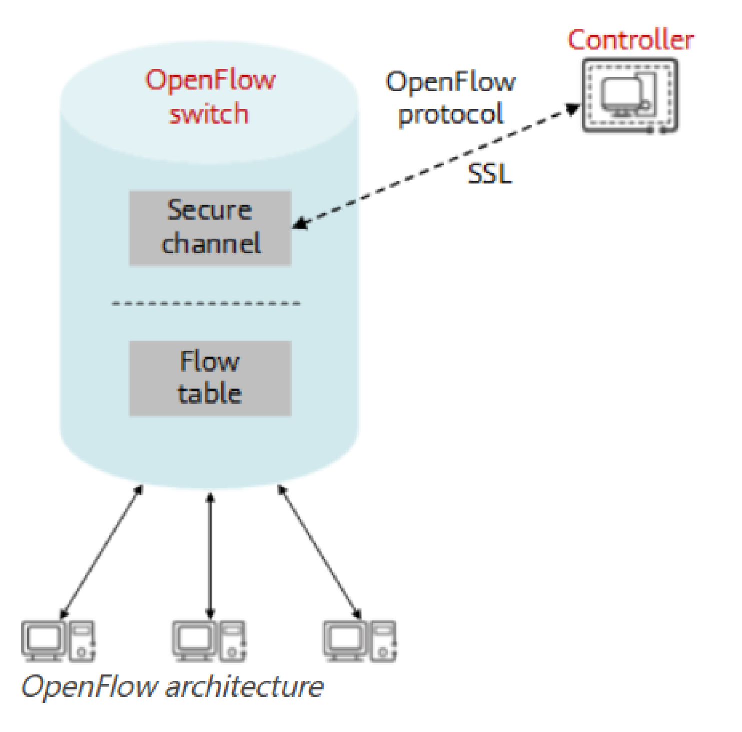
Dec. 2014

v1.5
- Egress table

Evolution and major changes of OpenFlow versions

How OpenFlow Works

- The OpenFlow architecture consists of a controller, OpenFlow switch, and secure channel.
- The controller controls the network in a centralized manner to implement the functions of the control layer.
- The OpenFlow switch is responsible for forwarding at the data layer; it exchanges messages with the controller through a secure channel to receive forwarding entries and report its status.



OpenFlow Controller

- An OpenFlow controller is the brain of the SDN architecture and is located at the control layer to instruct data forwarding through the OpenFlow protocol.
- Currently, mainstream OpenFlow controllers are classified into two types: open-source controllers and vendor-developed commercial controllers.
- The widely used open-source controllers include NOX, POX, and OpenDaylight

OpenFlow Secure Channel

- A secure channel is established between a controller and an OpenFlow switch. Through this channel, the controller controls and manages the switch, and receives feedback from the switch.
- The messages exchanged over the OpenFlow secure channel must comply with the format specified by the OpenFlow protocol.
- The OpenFlow secure channel is usually encrypted using Transport Layer Security (TLS), Controller-to-Switch message: is sent by the controller to the OpenFlow switch to manage or obtain the OpenFlow switch status.
- Asynchronous message: is sent by the OpenFlow switch to the controller to update network events or status changes to the controller.
- Symmetric message: is sent without solicitation by either the OpenFlow switch or the controller. It is mainly used to set up a connection and detect whether the peer is online.

OpenFlow Switch

- As a core component of the OpenFlow network, an OpenFlow switch is mainly responsible for forwarding at the data layer. It can be a physical or virtualized switch/router. OpenFlow switches are classified into the following types based on their support for OpenFlow
- Dedicated OpenFlow switch: is a standard OpenFlow device that supports only OpenFlow forwarding. The switch processes all traffic that passes through it in OpenFlow mode, and cannot perform Layer 2 or Layer 3 forwarding on the traffic.
- OpenFlow-compatible switch: supports both OpenFlow forwarding and Layer 2/3 forwarding. It is a commercial switch that supports OpenFlow features such as flow tables and secure channels.
- An OpenFlow switch forwards packets entering the switch based on the flow table, which contains a set of policy entries instructing the switch on how to process traffic. Flow entries are generated, maintained, and delivered by a controller.

Flow entry

- Traditional network devices such as switches and routers forward data based on the locally saved Layer 2 MAC address forwarding table, Layer 3 IP address routing table, and transport-layer port numbers.

- OpenFlow switches forward data based on flow tables that contain network configuration information of all layers on the network, instead of 5-tuple information.

- The entries in a flow table are flexible combinations of certain keywords and actions.

- Each flow entry in an OpenFlow flow table consists of match fields and a set of instructions applying to matching packets.

- When receiving a data packet, an OpenFlow switch parses and matches the packet header against match fields in the flow entries, and executes the corresponding instruction if a match is found.