

## ② SDN Control Plane Architecture. (Control plane functions)

- Now, the basic control plane functions,
- here in this fig, we can see different functions of the control plane.
- This fig shows the functions that are performed by the SDN controller.
- What are those functions?
  - Some of them are like, shortest path forwarding
    - ↳ It uses the routing information and that routing information is actually collected from the switches.
    - And based on that collected information, it establishes the preferred routes towards the destination.
    - That is the shortest path forwarding functionality and it is one of the key functionality of the control plane.
  - Notification Manager.
    - ↳ It receives, it processes and it forwards different events like, alarm notification like,

②

Security alarms notification or even changes in the stage of the switch for eg to an application.

↳ So Basically what is the main functionality here, it receives, it processes and it forwards events to an application. That event could be like, alarm notification, privacy, security notification, stage change and things like that. So it will notify the applications.

→ security mechanism.

↳ And here the focus is on security.

- It provides the security enforcement b/w the Applications and services.

→ Topology manager.

↳ It builds and maintain switch interconnection topology information.

→ Statistics manager.

↳ It collects the data on traffic through the switches.

→ Device manager.

↳ It configure the devices mean the switches, the different parameters of

the switches and the different attributes of the switches.

↳ And It also managed the flow table.

→ All these functions provided by the SDN controller.

- It can be viewed like an operating system.

- And in the context of SDN, that is known as network operating system (NOS).

- NOS provides essential services, common application APIs and also an abstraction of the low layer elements.

- These all functions enable the developers to define network policies and manage networks without concern for the details of the network device characteristics.  
Means → abstraction.

→ The northbound APIs, it enables the developer to create software, that are independent and not only independent from the Data Plane but also it provides a great extent of usability, that this developed SW could be able

or it can be used with a variety of SDN④  
controller.

(3)

## SDN controller Implementation

(1)

→ the first one in the list is the

OpenDaylight

→ The openDaylight is open source platform and it is founded by Cisco and IBM.

→ It can be implemented as a single centralized controller but it also enable controllers to be distributed more than one controllers are also able to work and collaborate in a distributed manner.

→ The second one is ONOS

Open Network Operating System

→ Open means a freely available - open source.

- It is an open source SDN network operating system.

- It was developed by a number of carriers like AT&T, TN, NTT and etc.

- The best thing about ONOS is that, it supported by ONF. It is going to be major factor in the SDN deployment.

(1) - It is used as a distributed controller and it also provide the abstraction functionalities. ~~that we just cover~~

## → POX

- It is also an OF - open source controller and it is implemented by a number of SDN developer and engineers.
- It provides the web-based graphical user interface and the best thing about POX is - it is written in python.

## → Beacon

- It is also an open source package developed at the University of stanford.
- It was the first controller that make it possible for beginner programmers to work with and create a working environment of SDN.
- This is like for learning purpose.

## → Floodlight

- Open source package and it was developed by big switch networks.
- Its development was based on beacon.
- But it has an active community and they have added a large number of features.
- - It has both web-based GUI, which is Java based and also it exposes the different functionality of the controller with the help of REST-APIs.
- - Here, we can use the REST-APIs at the northbound.
- - REST-API is actually exposed with the help of Floodlight.

## → Ryu and Onix

- Ryu is developed by entity lableg and it is fully developed in python.
- Onix is a distributed controller and it was jointly developed by VM Ware, google and entity.

And the downside of this last implementation  
is that, it is commercial.

⇒ In this list, the most significant  
controller is OpenDaylight and ONOS.

- The southbound interface is between the control plane and data plane.
- And the most popular API at this layer is the OF.
- This interface provides the logical connection b/w the SDN controller and the Data plane switches.
- here in this fig, we have southbound (next slide)  
(as<sup>33</sup>) interface shown by this double headed arrow and it provides the logical connection between the SDN controller and data plane switches.
- Some controller products and configuration they supports only a single southbound protocol. But on the other side, a more flexible approach is to use southbound abstraction layer in operating system.
- If you want to have a more flexible approach then you have to use a layer of abstraction and that layer of abstraction will provide

Common interface for the control plane  
functions and it will also support multiple  
Southbound APIs.

- Abstraction layer provides flexibility.
- The most common implemented Southbound API is OF. But besides OF we also have another API which is known as FCoEES.
  - Which stands for Forwarding and Control Element Separation (FCoEES).
  - We also have another one which is known as OVSDB - Open vSwitch Database Management Protocol and finally we have the 3rd one which is known as Protocol oblivious Forwarding (POF).
  - These are some of the APIs at the south bound.
  - But the most common one is the OF.

- OVSDB - it is used for virtual switching and that is interoperable with almost all popular hypervisors.
- It uses the OF for message forwarding in the control plane.

Other than that we have the Forces ③  
and it is developed by ietf (internet  
Engineering task force). They have  
standardized the interface b/w the  
control plane and data plane for  
internet protocol routers.

- third one is POF → is an enhanced version of OF.
  - It simplifies the logic.
  - It makes the logic very generic for forwarding the packets.

## Northbound Interface.

- It enables application to access control plane functions and services.
- This is possible without needing to know the details of the underlying network switches.
- This is viewed as a software APIs.
- The Northbound interface is viewed as an API rather than a protocol.
- The standardization is lacking at the northbound interface.
- There is no widely accepted standard for this interface.
- A number of unique APIs have been developed and these growing number of APIs making this complicated to develop SDN applications.
- To address this challenge the open network foundation formed the northbound interface working group.
- ONF has established NBI working group.

↓

Northbound Interface  
Working Group.

Their objective is to define and standardize a number of broadly useful APIs for the northbound.

- the most common one is known as a REST.

## (Geography Cally)

- this fig shows a simplified eg. of an architecture with multiple level of northbound APIs.
- Why we need these levels?
  - there is a discussion in the NBI working group about these different levels of APIs.
- Some APIs are further north while some are like also close to the controller - this is like a latitude.
- Where some of the APIs are close to the controller while other are far away from the controller.
- But they are close to the applications
- There are 3 APIs in northbound interface.

### ① Base controller function APIs

- It exposes the basic functions of the controller.
- It is closed to the controller.
- So it will expose the basic

unctions of the controller and can be used by developer to create network services.

## ② Network Service APIs.

- this APIs will expose the network services to the north.

## ③ NBI application APIs.

- These expose application related services.
- - Because It is close to the application plane.
  - so it will expose the application related services and those application are built on top of the network services.

# ITU-T Model

- This model is commonly referred by Y.3300.
- This is actually SDN high level architecture.
- It is defined in the ITU-T Y.3300.
- This model has 3 layers.
  - Application layer
  - SDN control layer
  - Resource layer.
- Application layer.
  - All the applications interact with the SDN controller with the help of API.
  - That form an application control interface.
  - The application make use of an abstracted view of the network resources and that view is provided by the SDN control plane.
- The control layer provides a means to dynamically control the behaviour of the network resources.
- That must be instructed by the

application layer.



- SDN is like the intermediate.
- It is instructed by the application layer and it manages and control the network resources based on the instruction that received from the application layer.
- This control layer can be viewed as having 3 sub layers. And those sub layers are Application support, Orchestration and Abstraction.

### ① Application Support.

- It provides an APIs for the SDN applications.
- As it is closest to the applications.
- So it will provide an API for these applications to access the network information.

### ② Orchestration.

- It provides the automated control and management of these network resources.
- It also provides the coordination of request from application layer for network resources.

The concept of orchestration is - it makes the automated configuration management and coordination of a computer resources.

### ③ Abstraction

- It is close to the data plane
- It will interact with the network resources.
- It will also provide an abstraction of the network resources for flexibility purposes.

⇒ The third layer of ITU-T model is the Resource layer.

- It consists of interconnected set of data plane forwarding elements ~~and~~ and those forwarding elements are commonly referred by switches.
- These switches perform the transport and processing of data packets according to the decision made by the SDN controller.
- Control support means it supports programmability of resource layer functions with the help of resource control interface.

- (4)
- And last we have the responsibility of the forwarding.
- for that we have the 2<sup>nd</sup> sub layer and that is Data Transport and processing.
  - That provide data forwarding and data routing functions.

## OpenDaylight

(4)

①

- This is for implementation of SDN Controller.
- One of the most dominant one is the OpenDaylight.
- It is an open source project and it is hosted by the Linux foundation and it also involves all major networking organizations.
- This project aims to produce an extensible, open source, virtual networking platform.
- And all of them on top of the existing OF.
- The OpenDaylight is to enable the industry participation or the industry partners to come together and to develop core open source modules.
- This is not a property of single organization but this have a consortium of different organizations. and they come together and work together and collaborate together towards a common opensource module.

It is important to understand OpenDaylight<sup>(2)</sup>  
and it will give you a good idea of  
the functionalities of a typical SDN  
controller because it is actually the  
implementation of the SDN controller.

# OpenDaylight Architecture

- This is the brief architecture of the OpenDaylight
- There are several components, some of them are quite familiar while some of them are new.
- The 1<sup>st</sup> one is
- This architecture consists of five layers.
  - At bottom we have the switches, the data plane.
- ① Networks applications, orchestration and services.
  - This layer control and monitor the network behavior.
  - It uses the controller together, network intelligence run different algorithms, perform different analytics and then use the controller to orchestrate the new rules.
  - And orchestration mean automatically implementation, configuration, monitoring etc.

## OpenDaylight APIs (REST, OSGi)

- OSGi Stands for Open Service Gateway initiative.
- It is like a set of common interfaces.
- OpenDaylight supports the OSGi framework
- The OSGi framework is basically used for applications that will run in the same address space as the controller means they are on the same address space.
- While that is not always the case, in some scenarios it is possible that, the application and controller, they are not in the same address space. They do not run in the same address space.
- Even they could be on a different machine, for that scenario you have to use the REST APIs.
- If they are in the same address space, same machine, then OSGi, if they are in the different address space then you have to use the REST APIs.

### → ③ Controller functions and services.

- This is the control plane function them and services.
- This is the SON controller.

### → ④ Service Abstraction layer (SAL)

- It provides the abstraction.
- It will provide a uniform view of the data plane resources.

→ The different resources are at Data plane.

→ SAL provides the abstraction.

- SAL provides the abstraction.
- The control plane function can be implemented independently of the specific interfaces and specific protocols and specific devices.

### → ⑤ South bound interfaces and protocols.

- It has support for OF and other standards protocols, vendor-specific interfaces.
- Other protocols like ONF, IETF.

- This OpenDaylight composed of both the control and application plane functionalities.
- This architecture contains both the control plane and the application plane functionalities. That is why this OpenDaylight is more than just an SDN controller.
- It has also support for other APIs.
- It not only tied to the OF or any other specific south bound interfaces.
- It will provide a greater flexibility in constructing SDN network configuration.
- The key element in this is the SAL.
- SAL allows to use multiple protocol on this south bound interfaces.