chap-5

Abstraction in SDN

— fig.

- SDN can be defined by 3 fundamental abstractions, forwarding abstraction, distribution abstraction and specification abstraction.

→ (1) Forwarding Abstraction

- It will allow a control program to specify data plane forwarding behaviour but it will hide the details of the underlying ~~switches~~ switching hardware.
- Why we need this abstraction? Because it provides the flexibility and vendor neutrality.
- We have different manufacturer devices at the physical level, so we need an abstraction, that our solution will work for all the diff^{n} vendor devices and that is known as vendor neutrality.
- this abstraction is normally implemented with the help of an API (Southbound API) and that is the OpenFlow.

→ (2) Distribution Abstraction

- This is b/w the application plane and the control plane.
- this concept arises in the context of distributed controllers.
- More than one SDN controller are possible for large enterprises, so a cooperating set

of distributed controllers, they maintains a state description of the network and routes through the network.
- So, if there are more than one controller, the must cooperate to maintain a consistent logical and global view of the whole network.
- But, this must be hidden from the upper layer.
- The objective of this abstraction is to hide complex distribution mechanism and due to this abstraction, we will have a single coherent global view of the network.
- And, the eg of this implementation is the northbound API, (REST). etc.,

→ ③ Specification Abstraction
- This will provide an abstract level view of the global network.
- It will provide enough detail for the application to specify their goals such as, routing policies, Security Policies, trust related policies, forwarding policies etc...
- Without providing the information, needed to implement the goals.

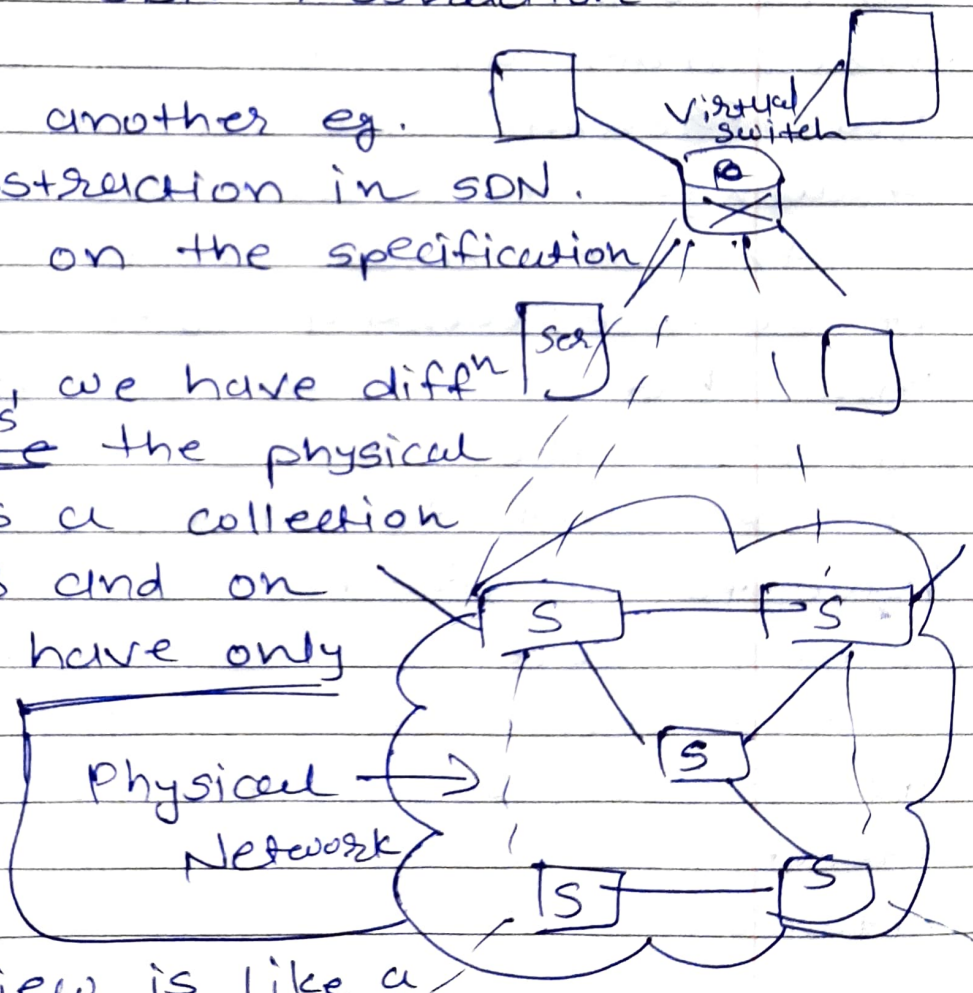
→ Forwarding abstraction hide the low level details, forwarding hw details from the higher layer.
→ Distribution abstraction hide the distributed nature of the SDN controller. It provides the global network view.
→ Specification abstraction will provide a network view to the application. A single network global

view to the application. It will hide the physical network details from it.

→ This figure is a simple example of these different types of specification.
- The physical network is a collection of interconnected SDN control plane.
- the abstract view is a single virtual network and at the lower level a single virtual switch.

(4) SDN - Abstraction.

- This is the another eg. of the abstraction in SDN.
- this fig. focuses on the specification abstraction.
- At the data layer, we have diff^n switches, ~~these are~~ this is the physical network which is a collection of diff^n switches and on top of that, we have only a single virtual switch. This is possible because of the abstraction.
- The abstract view is like a single virtual switch although we have multiple switches at a physical network. But the details are hidden.



(5) An example: Frenetic Architecture.

- This is the real example of network services abstraction layer. ~~from implementing~~
- Like the This is known as a frentic architecture.
- This is a programming language which enable different type of network related entities. like, network operators, network designer, policy makes, to program network as a whole instead of manually configuring these individual network elements.
- this Frenetic was initially designed to solve challenges that were related with the use of the OF based model.
- If we are working with OF, then we must know that it work at the southbound API, closer to the devices.
- If we want to directly work with OF, we have to go directly down to the network element level.
- But Frenetic includes a higher leves language. like the query language, that provides effective abstraction for reading the network states.
- It consists of two levels of abstraction, the upper level which is the source level API, the programming API. It provides a set of operators for manipulating the streams of network traffic.
- The lower level of abstraction is provided by a runtime system, that operates within the

SDN controller and it will translate high level policies and queries into low level flow rules and then it will issue the needed OF commands to the switches.

06 Traffic Engineering.

- Network engineering is a method for dynamically analyzing, for dynamically regulating and for predicting the behavior of data flow in networks.
- The objective of nework traffic engineering is to optimize the performance to meet the service level agreements and this traffic enginnering involve esteblishing routing and forwarding policies.
- Because it deals with traffic.
- This is something which is related to flow of the datel.
- Here we need policies, that are based on the QoS requirements.
- Within the SDN, the task of traffic engineering should be considerably simplified. Because we talked about abstraction. Abstraction should be included for the traffic engineering. as well.
- If SDN offers a uniform global and logical view of the network, of the switches, of the devices, of the physical layer entities.
- And it also provide very powerful tools for configuring and managing these network devices, the network switches.

- So the key idea is to simplify the task of traffic engineering.
- Some of the traffic engineering functions are already implemented in the SDN.
- These are functions, which focuses on the load balancing, Energy-aware routing & If you are going to ~~apply~~ deploy it in adhoc networks or in sensor networks, where energy is the main concern.) And optimization of resources with the help of scheduling mechanism, traffic engineering where the load on the network is going to be as minimum as possible, Recovery in case of failure, QoS enforcement and management.

(7) Policy Cop.

- This is the actual implementation of the Traffic Engineering.
- It focuses on the QoS policy enforcement.
- It provide a framework, where we can enforce different policies.
- The focus is on the policies, that is why it is known as policy cop.
- The key feature of this application is that it monitors the network, the underlying network to detect policy violations.
- Here we have a validator → policy validator.
- It will monitor the network for any violation of the network, of the policies.
- These policies are defined as SLS service level agreements.

- If the policies are violated, then it will reconfigure the network to reinforce the violated policies.
- This fig. consist of different modules.
- 11 modules and 2 databases. total we have 13.
- Every module is responsible for performing some functionalities.
- Here, 2 databases, which are related with policies with rules.
- These databases are installed, one Database is installed in application while the other one is installed in the control plane.
- This application is uses the control plane of SDN to monitor the compliance with quality of service policies. and then it can automatically adjust the control plane rules.
- Also, we have Flow Table concept here, Flow Tables in the data plane but that is going to be based on dynamic network traffic statistic.
- In control plane, the policy cop is relies on 4 different modules and one database.

① Admission Control → It will accept or it will reject request from the resource provisioning module for reserving network resources. If we want to reserve network resource, then the request must be generated and it is going to be passed through the admission Control mechanism. It decides based on the policies whether the request should be granted or rejected.

(2) Routing - It will determine the available path based on the control rules, that are available in the rule database.
- Rules for forwarding.

(3) Device Tracker - It will trace the updown status of these network switches.
- These devices needs to be monitored whether they are up or whether they are down. Their status must be shared with the application.

(4) Statistic Collector - It uses diff^n monitoring techniques to measure diff^n network metrics.

(5) Rule DB - The application plane will translate the high level policies to control rules and those rules are going to be stored in the Rule Database.

- Here we have a REST API, it connects this control plane module to the application plane modules.
- These modules need to communicate with one another and for that we have the REST APIs.
- Here on top of that, we have divided the modules into two parts, we have 2 modules having 4 and 3 submodules.
- The first one is the policy validator and the second one is the policy enforcer.
- A policy validator will monitor the network to detect policy violation and then a policy enforcer will adopt the control plane rules based on network condition and high level policies, It will enforce the

policies if violation occurs.
- Both of these modules rely on a centralized database.
- Policy database has access from the mgmt. plane, policy validator plane and also from the policy enforcement plane.
- Policy database contain QoS related policies, and these policies need to be entered to be defined by a network manager, network personal or a security related personal.

(1) Traffic Monitor — It will determine (33.11) appropriate monitoring intervals.

(2) Policy Checker - To check for the policy violation because, this sub module is for policy validation. We need to check whether the policy is defined by this person has been violated or not. And how it check for policies, from where it get the input, it gets its input from the policy database.

(3) Event Handler - It will examine violation events, if an event occurs then we need to inform those policies, so the event handler examine the violation event.
- It will then automatically invoke policy enforcer module by sending action requests to the network manager (Hey some policies were violated).

(4) Topology Manager - which deals with the topology. It maintain a global view of the network. or

(5) Resource Manager - It keeps track of the allocated resources.

(6) Policy Adaptation :- which need to be contacted by the event handler.

- It consists of functions, diff^n types of actions for policy violation, for eg :- if policy no 1 is violated then block the device, if policy no 2 is violated then remove the device.....

(7) Resource Provisioning :- It is directly connected to the database and this module either allocate resources or release resources based on the violation occur.

- Event handler inform the manager, policy adopter came into action and then it either remove that resource or do some action.

(8) Security

- Application this creed focuses on 2 things, or it has 2 objectives, (1) to address security concerns related to the use of SDN,
- Potential for new vectors of it attacks.
- like, there are diff^n types of threats for a network and that occur at the any of the level.
- The communication which is going on b/w any three layer (Data Plane, Control Plane, Application plane) that is also at risk because you cannot violate the security and privacy issues from any type of network.
- Security risks are there, which are related to the use of SDN and it could be related to the individual layers or it could be related to the communication that happens among

these different types of layers.
- SDN applications are needed to provide a secure use of the SDN itself that is the first goal. to address security concerns.
- The second goal is to use the functionalities of the SDN to improve the network security in which it is implemented.
- SDN presents a new security challenge for network designers, for managers, for researchers but it also provide a platform for implementation of centralize security policies and mechanism for the network.
- So the one side it is on risk but on the other side we can use the funcionalities of the SDN to improve our network security.
- Because SDN will allow the development of SDN security controllers and SDN related security applications that can provision security services and mechanism.

eg:- OpenDaylight DOS application
- It uses the existing SDN funcionalities to improve the network security.

(9) OpenDaylight DDoS Application.

- This is an open SDN security application and this is implemented within the OpenDaylight architecture.
- This is like a built-in component of this OpenDaylight project.
- The name is Distributed Denial of Service and this is a type of attack.
- Using OpenDaylight SDN controller, detection and mitigation of this attack is possible.

- This technique is uses a common approach to defense against the DDoS attacks.
- How it ~~could~~ deal with such type of attacks?
- It deals with this by Collecting Information about the traffic. Normal traffic baseline of the protected objects are built like, you record information about the normal traffic. and then in the second phase detection of the ~~distributed.~~ DDoS ~~attack~~ ~~patterns~~ are recorded. For eg:- If there is an attacks then the traffic is going to be different from the normal traffic.
- once the ~~A~~ attacks is ~~detected~~, the suspicious traffic can be blocked or they can be ~~diverted~~ diverted.
- It is like a 3 step process → (1) you collect statistic about the normal traffic (2) If there is a suspicious traffic pattern, you detect it (3) then you block it.
- This fig. shows a scenario.
- Here underlying SDN network consists of a number of Data Plane switches and these switches support traffic among different clients and server. On left side we have clients and on the right side we have servers and there is a traffic b/w the client and the servers.
- And that goes through the data planes switches.
- DDoS application works at the application layer ~~as~~ but it iteracts with the controller.

- Here we have Defence for All application, and interface which is based on the REST API and the user or the Network administrator or security person → He is able to use this application and it itself interact with the controller over OpenDaylight Controller.
- This application will make use of OpenDaylight Controller.
- This application support a user interface for the Network manager, that can either be a command line interface or it could be like, (REST based API).
- Here we have different components like the attack mitigation system, which is known as (AMS).
- This application will monitor the traffic of all of the switches. If the real time traffic exceed any % from the threshold defined then it is assume that an attack has been occured on the network.

(10) Big Data over SDN

- Our next application is the Big Data over SDN.
- Big Data is related to the cloud and with the data center.
- Cloud computing, Big data, large enterprise networks and information centric network they all strongly depend on highly scalable and efficient data centers.
- So cloud computing is related with data center, Big Data is related with data center.

- If there are large enterprises, they are going to use data center, even small enterprises are also shifting to the data centers concept.
- They have either their local data centers or they can purchase the services of the data centers by using utility model.
- So the key requirement for data centers is that, they are going to be highly available, their flexibility is going to be high, their latency is going to be low, their QoS is going to be high, their resource utilization is going to be intelligent, they need to reduce the energy consumption and they need to improve overall efficiency of the networks. These are the goals for the data centers.
- So, with traditional network architecture, many of these requirements, that are difficult to satisfy. That is why SDN offers a substantial improvment in the ability to rapidly modify data centers configurations, to ensure efficient operation of the network. means we need a way to shift data center from traditional network architecture towards the SDN based architecture.
- For that we are going see one example, the 1st one is related with the Big Data.
- Big Data is everywhere, now our problem is to use the SDN concept to optimize

Data center networking for Big Data applications, for that we need 3 things, we need to leverages the capabilities of the SDN, second point is - there is the characteristics of the structured Big Data, and the third point is the use of the optical circuits.

- Structure Big Data means, it depends on the existence of a mature data model, like the most traditional form of data storage, for eg:- excel file, SQL databases - where the data stores in tables, they are example of structured data. Because they have a matured data model. How the data organized, how the data is stored, how the data is accessed and so on---

- On the other side, unstructured data is information, that either does not have a predefined mature data model and is not organized at all. That is going to be very much difficult to understand this unstructured data by using traditional programs.

- the common egs are, If you have a data, that include audio files, video files and, non-sql databases etc--- So for to search information, that is going to be difficult and time consuming.

- So structured data is important, Specially for Big Data application.

→ The optical circuit is the third thing, this is like, related with the optical

transmission, like any piece of circuit switching equipment, that works b/w the fibers and why we use it? In order to get better control of physical infrastructure.
- This devices are going to reduce the multi-tenant data center configuration times. This is a concept which is related with the data communication.
- It will enhance the performance of the SDN.
- So, over all these 3 components are going to improve the application performance.
- This fig. shows simple architecture that is composed of SDN switches, But this is deployed in a data center concept.
- Here, OCS is the concept that, is circuit switch and O stands for optical,
- We have SDN controller, in this SDN controller is connected to 2 type of switches
- It is connected on the edge, to the OCS and on the other side, it is connected to the ethernet switch.
- All these switches are controlled by SDN Controller.
- It is also managing the forwarding b/w the switches, by using different forwarding rules.
- We have Hadoop - SDN controller is not only connected to switches but also to a new component, that is the Hadoop Scheduler. - It forms of queus of jobs to be scheduled. ~~And Hadoop itself is a huge topic.~~

- We also have another component, that is HBase master (Hadoop Based). This is going to be a master controller of a relational database. It will hold data for the Big Data applications. - SDN controller is also connected with the HBase.
- And for cluster purposes, we do have a Mesos cluster manager. It is an open source software package that provides scheduling and resource allocation services across distributed environment.
- So, for the centralized, we have HBase Master and for distributed we have Mesos Cluster Manager.
- If you want Big Data to be controlled, over SDN, this is a basic integrated approach.

(11) Cloud Networking over SDN

- To understand this application we need to see one framework that is CloudNaaS.
- It stands for Network is a service.
- In cloud computing, there are different models, like, application is a service, Platform as a service, Infrastructure as a service, Security as a service.....
- So, one of them is Naas. which means Network as a service.
- It is a cloud networking system, that exploit openFlow SDN capabilities to provide

a greater degree of control over cloud network functions by the cloud customers.

- It uses primitives that are directly implemented within the cloud infrastructure.
- So, this fig shows the sequence of events that happened in the framework.
- There are 4 sub images, each sub-image represent an event.
- (1) Specify user Requirements.
  - We have cloud customer and he uses a simplest policy language to specify network services, that are required by the customer application. And this policy statements are issued to a cloud controller.
- (2) Convert requirements into a communication matrix. (Communication b/w the cloud machines itself).
- Here we do not have any user.
- The cloud controller maps the policy entered by the user.
- Here, the main thing is mapping, so the cloud controller map the network policies entered by the user into a communication matrix. These matrix define the desired communication patterns and network services. Here policies are converted into a language that is understandable by the devices. The matrix is used to determine optical placement of these virtual machines on the cloud.

- The allocated resources is related to optical placement of the virtual machine on the cloud servers. This allocation is actually done based on the knowledge of other customer requirements because, the concept is known as multi-tenancy. The resources are not allocated to a single user, they are allocated to multiple users.
- So, there must be knowledge of who else is using those resources for efficient utilization.
- Then we have mapped the policies into matrix in the next action.
- (3) The next thing is to issue those policies. So, there is a logical communication, Matrix is translated into the network level directive for data plane forwarding.
- Here, we have the actual resources. The policies were issues by the customer, they were translated into the matrix and then they are forwarded as directive for the data plane forwarding elements → the switches.
- So, the customer virtual machines instances are deployed by creating and placing the specified number of virtual machines.
- Here in 3rd action, there are logical communications, that occurs b/w the matrix and the switches.
- (4) last scenario, in which the network level directives are installed into the network devices.

with the help of OF.
- The user (1st action) wants this last thing
- He want to communicate with network devices. But first policies were issued to the controller, then it will map into the network in to the communication matrix, then the communication matrix was translated into the network level directive. and then they were installed in the network devices with the help of OF.

(12) Cloud Networking over SDN (Another View)

(1:02)

- This fig provides an overview of the architecture that we have seen. (Cloud Nag
- Here we have 2 main components, that is Cloud Controller and network Controller at the top.
- The cloud controller provides a base, like we have infrastructure, the physical switches, so it provides the base for the Infrastructure as a service.
- The user can communicate or can send request for setting up these virtual machine instances.
- He can send with cloud controller.
- Here cloud controller is managing the software,
- Cloud Controller construct the communication Matrix then translate to the next controller.

- The network controller uses the communication matrix to configure data planes for the switches.
- This is how the Naus architecture works.

(13) Information Centric Networking

- The Information centric Network sometimes known as CCN - Content Centric Networking.
- It received attention of the ~~reche~~ researchers in recent years.
- The basic concept is that, we have a traditional host centric network paradigm and now we have an information-centric network paradigm.
- In the traditional host centric network paradigm, information is obtained by contacting a specified named host, like, we have the IP addresses ~~g~~ we have the Port addresses and things like that.
- The basic idea of ICN is to decouple information from its sources.
- Information sources can place an information, user can find information anywhere in the network, because the information is named, addressed and match independently.
- So the focus is on information not on the specific IDs. Instead of specifying a source, destination, host pair for communication

- A piece of information itself is named.
- So, when a request is sent, the network is responsible for locating the best source of information, that can provide the d

Chap - 5
SDN control Plane

→ Control layer is the second layer in the SDN architecture.

→ In that, there are different APIs and diff^n functionalities.

① SDN Control plane Architecture.

⟹ The main component of SDN control architecture is, it's actually maps application layer services requests into specific commands, As we can see in this SDN architecture → the main functionality of control layer is to mapping. Mapping blw the Applications and converting a different service request to commands. And that commands are passed to the data plane switches, That is from the one side, from the other side, it supplies applications with information about the data plane topology and other activities.

- It is important to remember that, control plane is implemented as a server. or if there are more than one SDN controller, then they are implemented

as a set of servers and these servers are known as SDN controller.

- So basically, the SDN controller are servers. It could be a single controller or it could be a set of controllers. If it is a set of controller or set of servers, then co-ordination among these different servers or SDN controller is required.