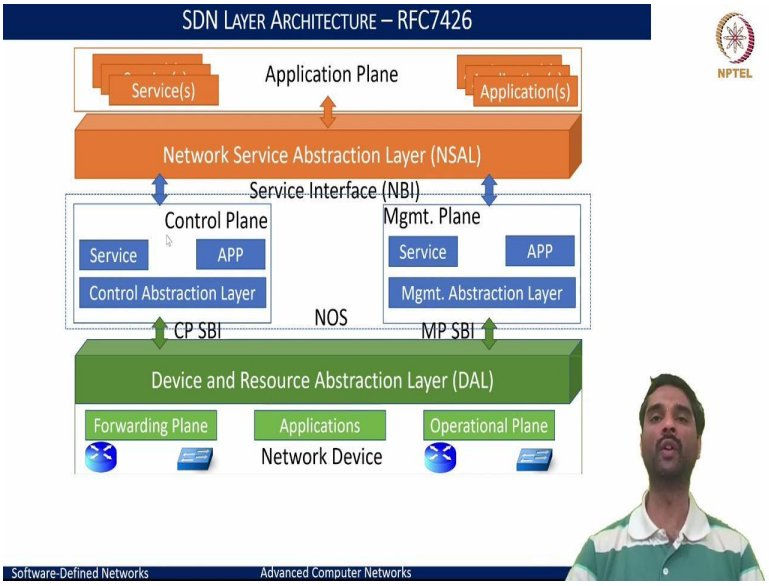
Advanced Computer Networks
Dr Sameer Kulkarni
Department of Computer Science Engineering
Indian Institute of Technology, Gandhinagar
Software Defined Networking – Part 3

(Refer Slide Time: 0:17)



In this lecture, let us try to understand how SDN evolved, what are the key layers and the components of each layer and understand the whole essence of how SDN looks now from what it was in the earlier days. So, at the bottom of the SDN stack, what we see is a network device. Note this network device could either be physical or virtual. What I mean is the routers and switches that we are considering could either be physical devices, or they could also be softwarized virtual devices that would be running.

And in the view of SDN architecture, these network devices consist of two key planes one, the data plane, also known as the forwarding plane, and the other, the operational plane; what we mean by them is basically the forwarding plane is the one that is responsible for handling the packets in the data path that is based on the instructions that it would receive from the control plane and take necessary actions to forward the packets that it receives, it may not necessarily be limited to forwarding, but it could also be to drop the packets, to send the packets to the control plane itself or even to take some actions such as modification of specific packet headers. And overall, this is basically the transformation of the packets and is a termination point for the

control plane services that would act on these packets. And even if we think of like classifications, they can be handled in the forwarding plane.

And the other part is basically the operational plane which is responsible for managing the operational state of the network devices. In essence, if we have a switch that has 12 ports, 24 ports, what is the status of each port? How do you configure the VLAN IDs for different ports and whether these ports have to be active or inactive, and management of the device, in essence, is what constitutes this operational plane. And this is usually the termination point for the management plane of the network operating system that would communicate and coordinate to set up the specific characteristics of the device.

And many times, there is an ambiguity between operational and forwarding planes. But we can confine to the understanding that the forwarding plane is anything to do with the processing of the packets that the device has to do, and the operational plan is with respect to the internal management of the device in itself, be it a physical or a virtual device. Nonetheless, there could be packets that could come for operational plane for example, SNMP, when you want to transmit the data, there may be some operational packets that are being processed for this case.

And the applications in large constitute the functionalities for the forwarding plane operations and operational plane operations within these networking device, and this is the model of any switch or a router that we consider. But now, with open API's that we want to bring in the SDN layer, what is more crucial is to add a device and resource abstraction layer. And this is the layer that is responsible for exporting out the key interfaces through which the network operating system can communicate either with the forwarding plane of the device or with the operational plane of this physical or virtual router and switch device.

And on top of this device and resource abstraction layer would be our network operating system or what we call it as a remote SDN controller. And now, this SDN controller would constitute of two key components, one is the control plane, which is responsible to communicate with the forwarding plane or the data plane of the device and essentially ensure all the configurations of the flow rules that would affect the way the data packets will be forwarded on this device.

And this control plane consists of the service layer in terms of what key services that this control plane is going to provide, and the control plane applications themselves and both can leverage

what we call the control abstraction layer in a way that you are trying to build the module that is able to make the communications for the respective southbound interface that is the control plane southbound interface to communicate with the device and resource abstraction layer.

And the most typical model that we are going to learn is about the OpenFlow interface for this control plane to communicate with the forwarding plane of the devices to set the forwarding rules, and the other plane is the management plane within the network operating system. And likewise, control plane, it would have its own core services and applications and it would use the management abstraction layer and the management plane's southbound interface to communicate with the operational plane of the device. And most often, this management plane's southbound interface could be like SNMP or NETCNOF and YANG and there have been variants that have evolved.

But our focus would primarily be on the control plane aspects as we go forward. But, in a nutshell, this network operating system would consist of both the control plane which is responsible for making the decisions on how the packets would be forwarded from one network device to the other, or any manipulations of the data or packets that have been received by a particular device. And the management plane is responsible basically to monitor and configure the maintenance aspects of the device, including the management of configuring the forwarding plane aspects like what VLAN IDs to push when the packets are received from a particular port and so on.

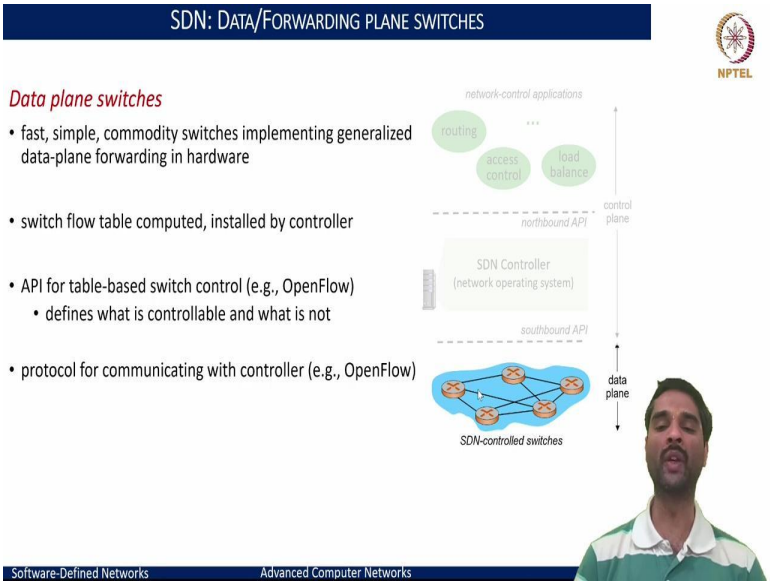
And this network operating system would then provide what we call a network service abstraction layer to ensure that we can build a very rich set of networking applications that can take use of both the control plane and management plane aspects of this network operating system. So, this network service abstraction layer plays a key role in terms of how the networking applications would then communicate with the control plane and management plane to ensure the intent of the network operators is being ensured on the networking devices.

And on top of this network service abstraction layer could be the application plane, where eventually, all the applications and services that define the network behavior would reside. These applications support the operations for forwarding plane, and the applications may be implemented in either modular as well as distributed fashion. For example, traffic engineering

that we spoke about in the earlier lectures, if we want to collect information about all the networking devices, the application plane may be distributed across multiple network OS to capture different specific aspects of monitoring the health of network devices, statistics of how many packets were processed, how many packets were dropped at each of the routers and devices and so on.

And like even we can think of NAT as an application, MPLS as an application that could run utilize the control plane services to actually configure these rules onto these network devices. So, that is, all planes mentioned above could be connected through these interfaces. And what we see at a part is below the network operating system is the southbound interface, and above the network operating system is the northbound interface. So, by southbound what we mean is a communication to the network device, either virtual or physical, and northbound interface is communication with respect to the network intent, so the applications that we want to run and make the adaptation onto these network devices.

(Refer Slide Time: 9:11)



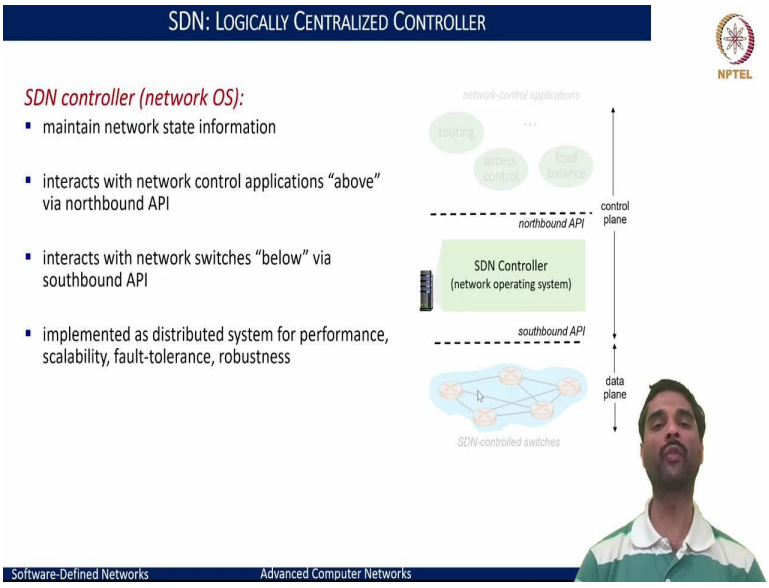
Now, if we look at each of these layers and try to understand the core functionalities and aspects of how the SDN layering is built, let us start to look at the ground-up fashion. The data plane switches are primarily the lowest end of the components, which basically constitute either the physical or virtual switches and routers that are connected in a network and controlled by the SDN controller. By this, what we mean is these data plane switches would be fast, simple

commodity switches that could be implemented on a generalized data plane forwarding in hardware.

In essence, these can not necessarily be just the custom switches that you make, but also the Linux servers or machines that we can transform into routers and switches. And what these really do is provide a means to set up the flow tables based on which the data will be processed on these devices and forwarded, and these flow tables would be computed by the SDN controller above. And it would then inject these flow rules into the tables of each of these network elements through the use of southbound API, necessarily the OpenFlow being one of them, but not necessarily just the OpenFlow, it could be any other API as such.

And this dictates how the SDN controller would then be able to communicate and the protocol that facilitates this is what we are going to learn is the OpenFlow API's.

(Refer Slide Time: 10:48)

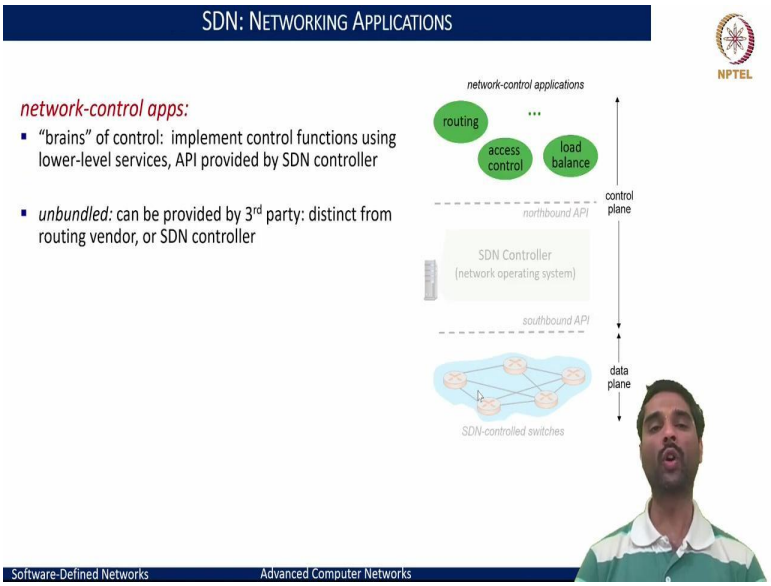


And on top of it is the SDN controller or the network operating system, as we referred, and this is the key piece which is going to maintain the network state information. And it is going to collect this network state information through the southbound API and communications with each of the networking devices that it is tasked to govern. That means it would interact with the network control plane or forwarding plane and the management plane of these devices, and extract specific data and maintain this information. And through this information, it would export

specific API's to the applications above so that they can all leverage this topology information, the statistics, and the status information and build necessary applications.

And this SDN controller is not necessarily just a centralized single entity, it could be distributed for the sake of guaranteeing the performance and scalability aspects, including support for failure, resilience, and providing robustness. So, we can think of this SDN controller as a distributed network operating system. And hence, we termed this as a logically centralized controller, which could be physically distributed.

(Refer Slide Time: 12:16)



And on top of this SDN controller would reside the core network control apps. And what we mean these network control applications are in fact, the intents that we want the network devices underlying network devices to do and honor. And by this, we mean routing, which is a fundamental aspect, when we connect multiple of the routers becomes an application that is programmed. And operator controls exactly how you would want to do the routing for each of these devices for each of the flows.

Access control in terms of the data plane or the control plane access controls that you would want to define what packets need to be allowed, not to be allowed. All of these can be configured through an application of access control, load balancing, what we spoke about in the earlier session of what was not possible with traditional networking. Now, we can define exactly the key

rules or the principles on which we would want to balance the load across a given network. We could even partition this network into isolation, providing different characteristics for the throughput bound flows versus the latency bound flows.

All of this can be managed as a program now in the network application plane. And note that when we say that we can have these network applications running on top of the SDN controller or a network operating system, we may have a variety of network operating systems built by different people. Some could be open, some could be proprietary, but what we would want is also to leverage these network applications to run on top of different network operating systems or different SDN controllers.

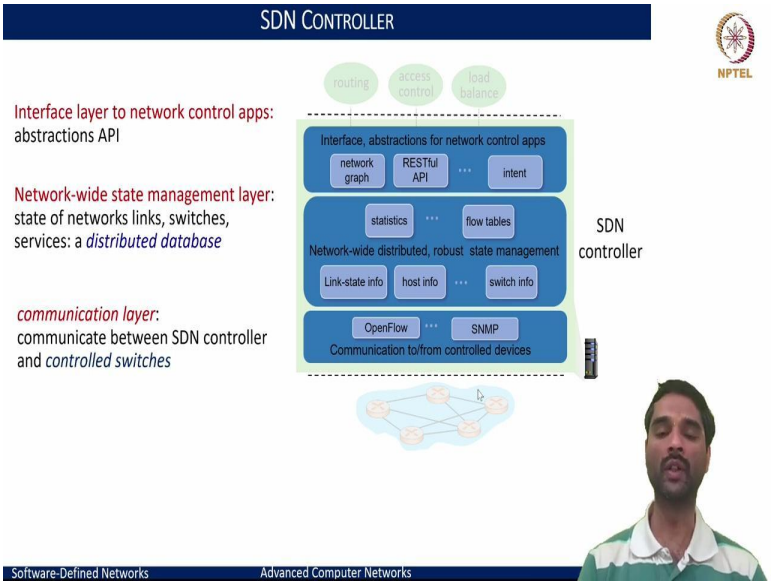
And in another way, we would also want these network control applications to be able to run not necessarily for just one particular SDN controller but for various other SDN controllers. But it is natural now to think of SDN controllers, the vendors of SDN controllers to provide the default network control applications, just as you see the Windows or Linux come with the default apps. But then, it should not confine Windows to just the Windows applications. Any third party should be able to develop the applications for Windows. Likewise, like for our mobile phones, Android as an OS, and you have the application space where any third party will develop the applications.

In the same sense, if we think of SDN controller as a network operating system, whoever is a vendor of SDN controller would provide certain default applications. But it would also open up for any of the third party to provide the necessary applications. That way, the innovations are propelled as to the greatest extent. And we can try out various things and very easily by building different network applications for a particular controller and trying it out on any of the networking devices. So, in essence, this also decouples, the application provider or in fact, if you put the researchers in the space, researchers would now be decoupled from worrying about what are the hardware characteristics through which we need to develop the network applications.

Do I need to look at the spec sheet of a Cisco router, or Arista router and see how I should build the application? Now, that is not necessary we just look at the northbound API's that are being provided by a specific network operating systems that we want to use, and then build the

applications by leveraging the services provided by that SDN controller. This, in a way really greatly simplifies what we can think of networks becoming much more agile and robust.

(Refer Slide Time: 16:14)



So, let us look at some of the SDN controllers. And in general, what it would encompass and few of the most well-known SDN controllers. When we think of SDN controllers, or a network operating system, think of it as two specific layers, just as any operating system, it would abstract out the hardware details, and in this network operating system, it would abstract out the details and characteristics of the networking devices. But nonetheless, as an OS, it would cover and provide the layers or interfaces to communicate and configure these devices. And that being OpenFlow for the control plane aspects SNMP, NETCONF for YANG for the management plane aspects, that becomes like the bottommost layer into the network operating systems. Very similar, if you think of the operating systems, the device drivers being at the bottom end, and then the core of the SDN controller is to manage the state and provide the key services up to the applications. And in order to do that, this would constitute building the network wide state, maintaining the information about the devices, maintaining the information about the necessary hosts that would be operational and constituting, maintaining the information about each of the links and statistics, each of the flow statistics and so on.

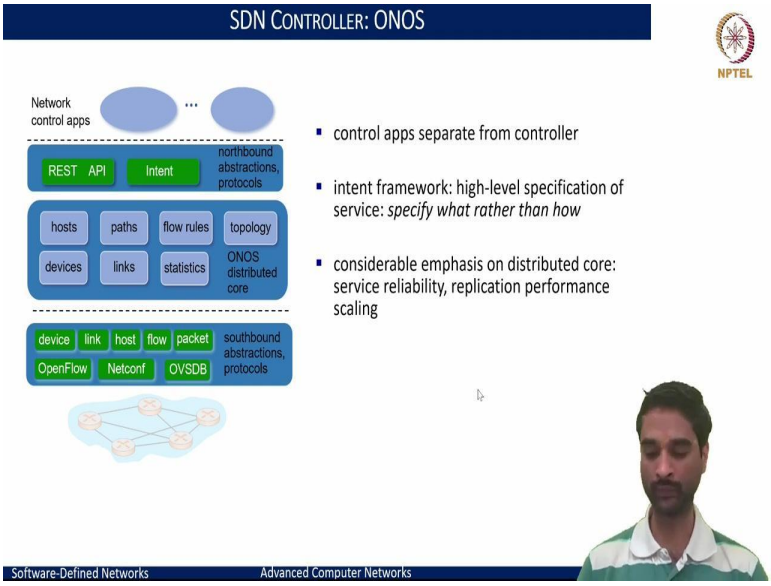
So, this core is what would be leveraged as services to be provided to the applications. And in essence, this is basically the metadata that the SDN controller would operate on. And on top of this metadata is what you want to present the abstractions to the applications that you want to run on top of this SDN controller. And for that, the most fundamental is the network graph abstraction, which gives the information of what nodes are connected with what ports, what the topology looks like, and then the API model that can enable applications to configure these devices.

And typically, this would be like a RESTful API, so that you want to be as stateless as possible so that it is easier to configure the content and think of the communication channel as a pipe. And then it could also provide the abstractions for the applications to instead specify intent than actually program the actual devices. By intent, we can think of very high-level languages, like in XML, how you would type and classify the struct and say what operation you would want to build. A very high-level user language-based notion can be used as an intent to say, load balance for these particular flows on these particular devices, that as an intent can then be translated all the way down to say, you have to configure router 1 with these flow tables, router 2 with these flow tables, and so on.

So, that way, it makes the application space lot more easier to configure and work with. And all the way what this SDN controller would be is sandwiched between two interfaces, that is northbound interface, which exports the API's that the network applications can use, and the southbound interface, which provides the means for the SDN controller to configure and control the underlying hardware, network devices. It could be hardware as well as virtual devices. So, we looked at the key communication layer, which is between the SDN controller and the control switches.

And I spoke about how this network-wide state management could be done. And in fact, here you can think of having these states set up as a distributed database, wherein multiple of these network operating systems can go and update on the same database to have a coherent view of the state. And the interface above is exactly the abstractions for the applications to be built.

(Refer Slide Time: 20:33)

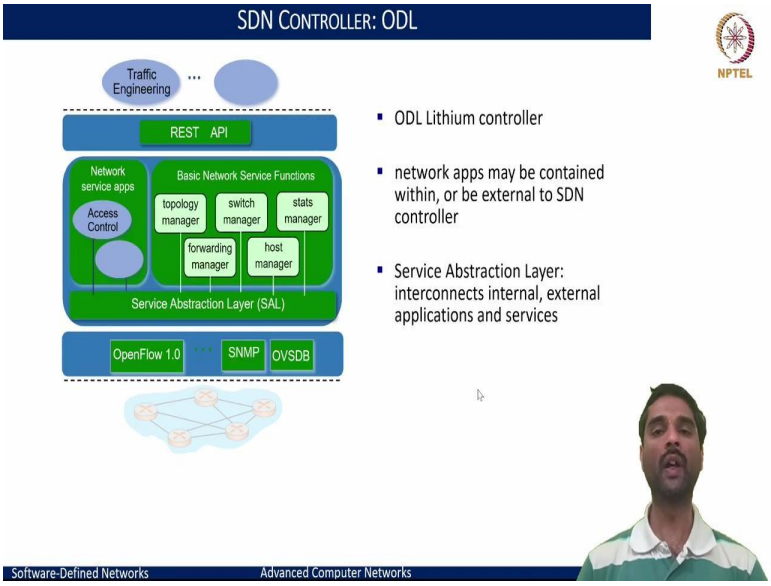


And the most popular SDN controller nowadays is the ONOS open network operating system. And here, it aligns with what we just discussed, but only that we need to remember that the intents are now whatever that are shown here, especially REST API are the abstractions that are provided as a northbound interface to the application. So, when I write an application, I have a choice of defining the owner's intent, and then pass the information down to the devices.

And on the southbound, the ONOS, supports OpenFlow for the control plane support to configure the forwarding plane of the data. And NETCONF as a means for the management plane, or the MPI the management southbound interface to communicate with the devices and configure the devices. And as the ONOS being a distributed core itself, it provides the information about the hosts, the paths that are available, the flow rules that are being applied, the topology information, and the statistics governing to each of the devices in terms of port statistics, flow statistics, all of these as services to the higher applications.

And the intent is a key aspect here because it is a very high-level specification, where you would just help the network operator communicate what he wants to do. And then the intent can then be programmatically assembled and understood by the network operating system, how things can be achieved. And what are the operations that it needs to do in a very transparent manner. So, that it really levies off the burden on the network operator to understand the how aspect of dealing with the networking devices.

(Refer Slide Time: 22:31)



And the other important SDN controller is the OpenDaylight or the ODL controller, and you will see that it looks more or less very similar. There have been various variants of this ODL controller that have evolved, but what it primarily tries to do is provide the REST API's for the applications as a northbound interface. And at the bottom, it provides the service abstraction layer for different networks for service functions that it builds. And this service abstraction layer would then communicate with underlying devices, either using OpenFlow, or SNMP for the control plane to data forwarding plane updates or management plane to the operational plane updates on to the devices.

And OVSDB is the database for the open switch perspective in terms of what information are being set and updated to configure the database rules onto the forwarding plane of the devices.