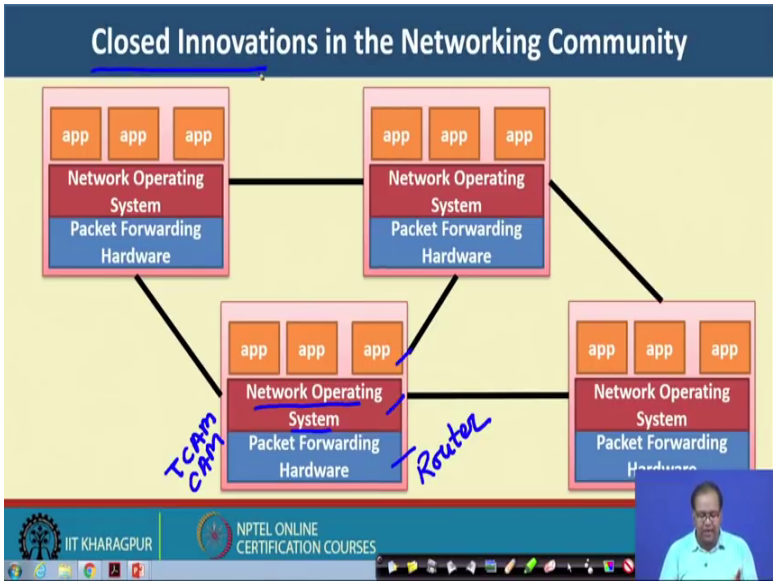
Computer Networks and Internet Protocol
Prof. Sandip Chakraborty
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 44
Software Defined Networking – II (OpenFlow)

Welcome back to the course on Computer Network and Internet Protocols. So, in the last class, we were discussing about the concept of software defined networking and we have looked into the basic architecture for the software defined networking and the broad concepts around that. So today, we will gradually go towards an implementation perspective of the software defined networking. And the software component of it, where the routers are the network devices are software controls.

So, we look into that what type of software or which software will actually control this entire networking architecture. So, in this context, we will look into a specific open source protocol implementation, which is called open flow and now subsequent classes, we look into a demonstration of ISD network by utilizing open flow protocol stack.

(Refer Slide Time: 01:16)



So, let us go to the details of that. So, as you are discussing that traditionally, in the network community, the innovation is a closed innovation. So, closed innovation means in a single box, we have the individual components or individual networking components, which should be there like here, if you just follow this diagram you have

the packet forwarding hardware, the network operating system and the different kind of networking application everything is bounded on the single bounding box. So, this contains the router hardware so, you can talk it as a router.

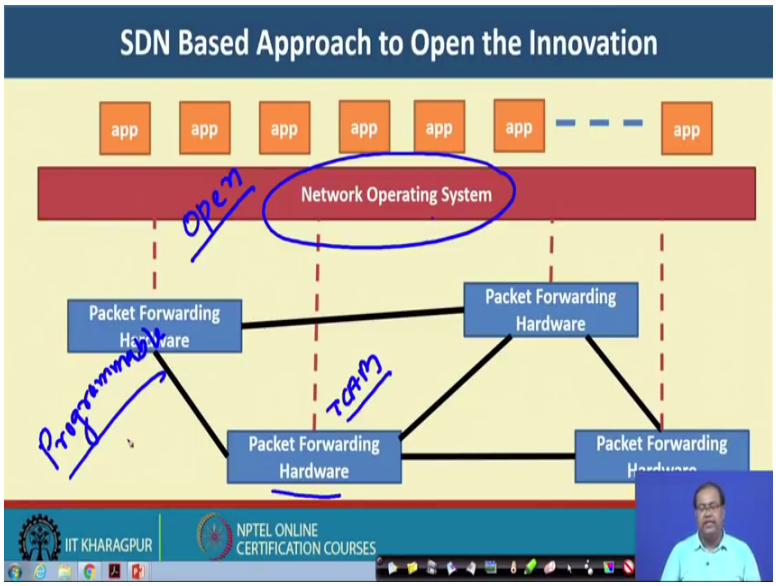
So, this contains the router hardware and inside the router, you have the hardware component, which is implemented in T CAM or in traditional router, it is CAM. So, one of this type of hardwares on top of that, we have the network operating system that actually, implements the different kind of networking protocols and routing control protocols. And on the top of that we have different kind of applications like the firewall application, the packet forwarding applications, which are there inside a router. Now whenever, you purchase a single router, this entire bounding box that comes from a vendor and that is why, we call it as a closed innovation.

So, closed innovation basically, says that well both the hardware as well as software along with the applications which are associated with your network functionalities, everything is coming from the same vendor as the same packets as a result the problem comes, in terms of first interoperability, the second tester network manageability. So, if you purchase a Cisco router and a Netgear router or a HP router. In general, it is the burden of the network operator to look into the individual configurations of the routers and then configure them in such a way so that, these routers can talk with each other or forward packets from one router to another router

And these operating systems, they being the vendor specific like whenever, you are purchasing a router from Cisco, the network operating system is coming from Cisco, Cisco internet operating system, IOS or any other operating system, whenever you are purchasing a router from Netgear, the operating system is coming from them.

So, because the operating system and the farmer are hardware dependent and it is vendor specific and that is why, the configuration or handling the operating system for a network administrator, who will actually give the input of the networking policy, for them this entire architecture becomes difficult to handle. So, because of that we gradually move from this kind of closed innovation network to a open innovation network.

(Refer Slide Time: 04:24)



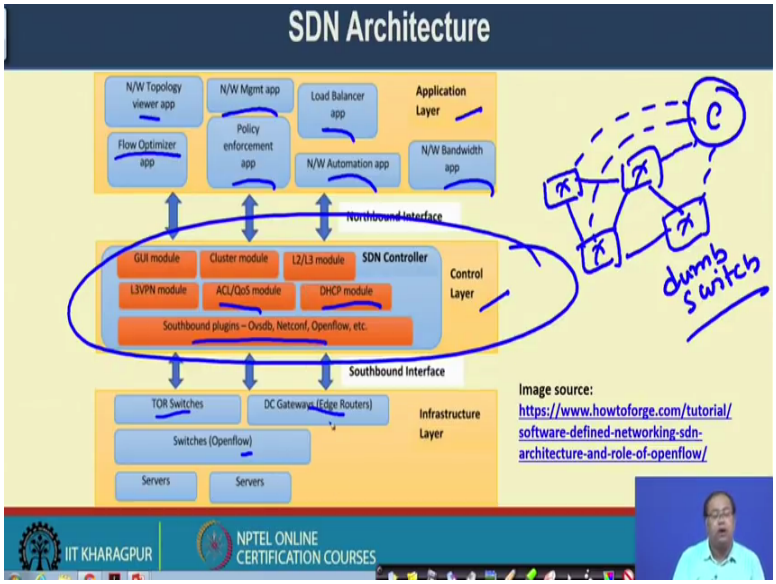
So, the open innovation network basically talks about something like this, now, the vendors will only supply the packet forwarding hardware, which at the dumb switches or we call it as the open switches or blind switches. So, there are different terminologies, which are being used in the networking community. So, you are getting this packet forwarding hardwares from different vendors and then you have a central network wide operating system, this network operating system, which is a open operating system.

So now, we are basically, segregating the hardware from the software or in routing functionalities as we are discussing in the last class, we are separating out the data functionalities or the data part functionalities from the control functionalities. So, the data for that, functionalities are still implemented inside the hardware inside this packet forwarding engine, where you have this TCAM implementation.

So, it is still in the hardware side, but on the software side, you can choose your own software, which you can utilize to configure all these hardwares all together and then on top of that you can write your own network applications and the advantage is that now, you do not need to bother about, whether your routers are coming from Cisco or from Netgear. You can purchase the hardwares from any vendors in the world and then you can connect it in your network and all these hardwares should be programmable with the help of this network operating system. So, the only modification in the hardware side is that. Now this, hardwares the packet forwarding hardwares are programmable hardware.

So, these are programmable hardware; that means, with the help of this network operating system, you can dynamically program the hardware so, that you can install the forwarding rules or you can install the configurations dynamically, based on the network operating system that you are using. So, the question comes that with this kind of SDN architecture, where we are gradually moving from a closed innovation network to a open innovation network, what are the requirements and how will you fulfill those requirements?

(Refer Slide Time: 06:48)



So, the things come from here. So, this was the brought SDN architecture as we are discussing. So, we have the infrastructure layer at the bottom. So, this infrastructure layer have different kind of hardwares like, you have the servers, you can have the open switches, you can have the top of rack switches, you can have edge routers or datacenter gateways. Then, we have a control layer in between and that control layer has all the control functionalities, which are inbuilt inside that. And on top of that, you have different kind of networking application. The application layer, where you can write down the network application like, the flow optimizer, the network topology viewer, network management application, policy enforcement application, load balancer network, automation network, bandwidth management, whatever application you can think of and you can design on top of that.

So, this control layer, which works like a brain of this entire network, we are making it logically centralized. So now, the architecture is that we have multiple hardware components, which are there and all these hardware components are connected to a central controller. And this controller actually, contains the control layer. So, these hardwares are connected over the network as they are there in the normal network. So, because these are just a blanket hardware or this does not have any kind of software inbuilt, we call it as the blind hardware or the open hardware and in case of normal networking terminology rather than calling it as a router, we call it just like dumb switch.

So, this dumb switch does not have any knowledge about what to do. Neither layer 2 knowledge, non layer 3 knowledge and this controller, whenever you are programming this controller, this controller will actually configure this dumb switches is dynamically, put the intelligence inside the switch. Now this controller has all these different kind of module, which can be there like you can have a GUI module, the cluster module to form a cluster, the layer 2, layer 3 module to implement the layer 2 and layer 3 functionalities of the network protocol stack, the VPN module to create a virtual private network. The quality of service module or access control list module to implement access control list or quality of service, DHCP module to implement DHCP protocol and the plug INS, which he called normally as the southbound plug INS to interact with the programmable infrastructure layer.

So here, you have 2 different interfaces from the control layer. From the control layer one interface that talk with the applications different kind of applications that, we call as the northbound interface and then with the control layer to the infrastructure layer, we require another interface, which is called a southbound in process. Now the task of the northbound interface is to understand this individual application layer program, and these application layers programs are implemented by utilizing your favorite programming language. So, you can use Python, you can use Java accordingly, you have to choose the controller.

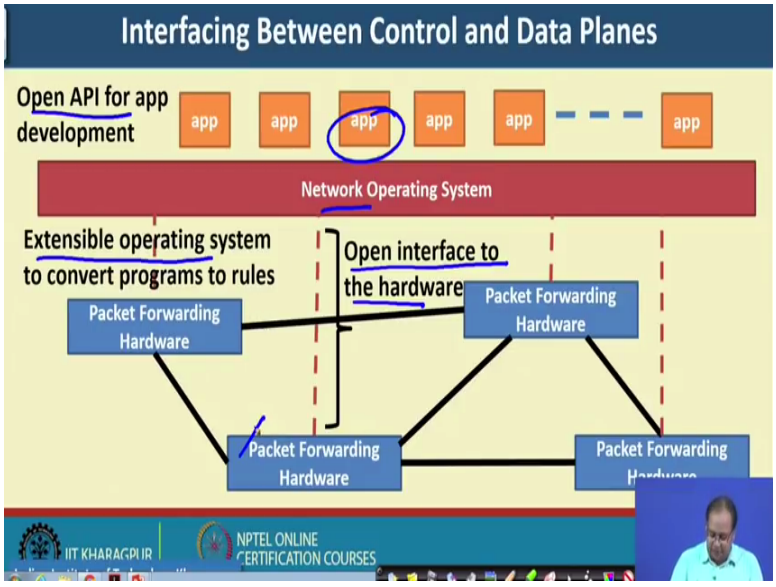
So, for example, in a typical SDN at work if you are familiar with Python, you can choose the POX or you kind of controller, if you are familiar with Java programming language, if you want to implement your application using Java programming language, you can use open daylight controller with support Java programming language. So, you can write down this application with your favorite programming language and the task of

this northbound interface is to understand, what is written inside the application, compile the application to the corresponding network protocol. And then map it to one of these modules, which are there inside the controller.

So, you can utilize this module to write your own program say for example with the help of you can write a load balancer application, where you can utilize the layer 2 layer 3 forwarding module to forward the packet to a specific destination. Now this control layer from these individual network functionalities, they have to convert it to the rules, the rules that will be programmed to the router and whenever incoming packet comes that rule will be executed.

So, from this individual network protocol to the rule conversion, that is done by the southbound interface. So, we look into the details about, how these rules are being implemented, and how you convert a particular protocol to a corresponding rules in the HT and terminology or in the open flow terminology.

(Refer Slide Time: 11:52)

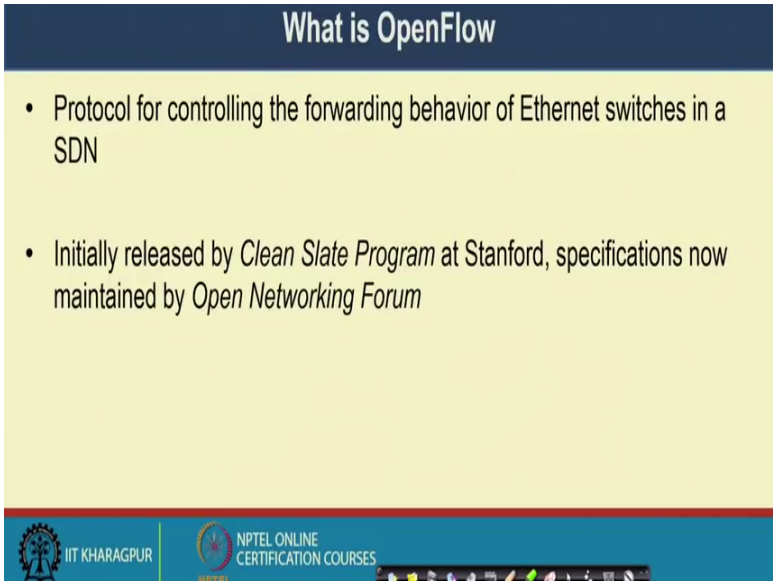


So, what we require in summary? So, to talk between the network operating system and the corresponding infrastructure, we require an open interface to the hardware.

So, that you do not depend on the corresponding vendors to program your network we require a open interface to the hardware for that. The second thing is that we require an open API for the application development so, that any application developer can develop

a network application. And the third thing is that, we require an extensible operating system to convert the programs to the routes. So, these applications they are nothing but a program from that program, we need to map it to the corresponding rule, which will be executed at that TCAM hardware of the packet forwarding engine, which is their inside the switches.

(Refer Slide Time: 12:51)



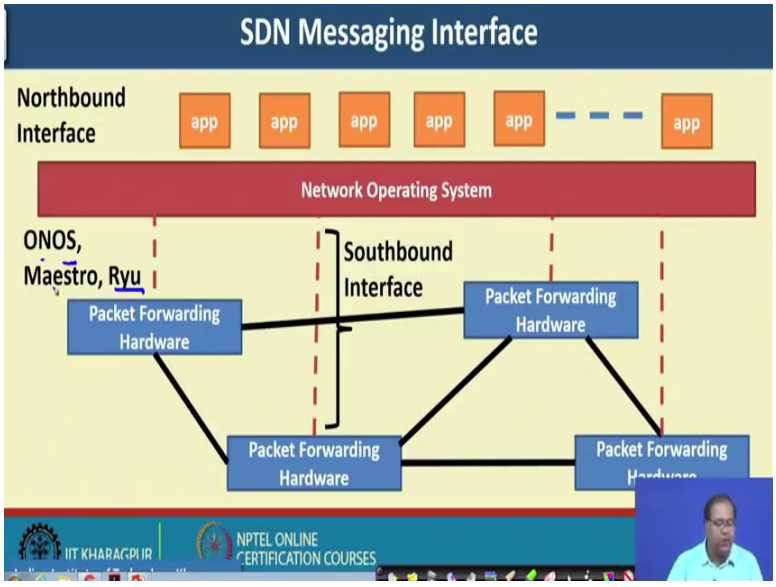
Well. So, what is open flow? So, open flow is a protocol for controlling the forwarding behavior of Ethernet switches in SDN network. Initially, this concept of open flow was released by clean slate program at Stanford and currently the specifications are maintained by open networking forum. So, the interesting fact is here that now this, inter-networking architecture, they are becoming open, they are moving from a closed community or from a vendor specific community to a open networking community, where every vendors join all together.

So, the vendors are building their hardwares and the community is building the open source operating system and the interface to interact with the corresponding hardware. So, this helps in 2 ways first of all, it makes the innovation specific or it makes the innovation rapid because now this entire software is open to the community, you can design your own network protocol and taste on a hardware for that, you do not need to search for an hardware of where, you can implement your own protocol, you can purchase any SDN supported switch, open switch. And then you can do your protocol

implementation on top of this open source operating system and the second advantage comes from the network management perspective, where the network administrator, they do not bother about reading the 1000 page manuals for from 3 different vendors.

So, they can just concentrate on a specific operating system and then try to write their own rules on top of that specific network operating system.

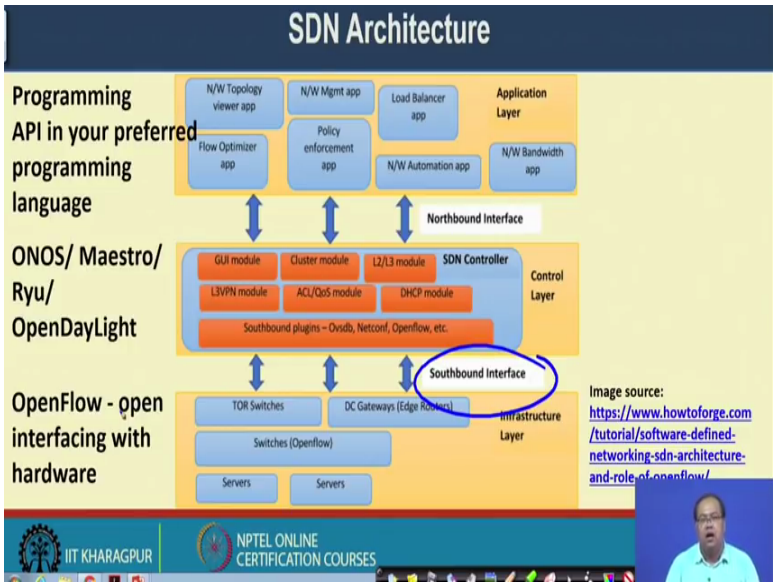
(Refer Slide Time: 14:43)



So, in terms of SDN messaging interface as we are mentioning that from networking operating system to the hardware, we have the southbound interface. Then from the network operating system to the application, we have the northbound interface that provides the programming API. And then this network operating system can be implemented in any open source operating system, there are multiple standard industry, specific operating systems, which are available nowadays.

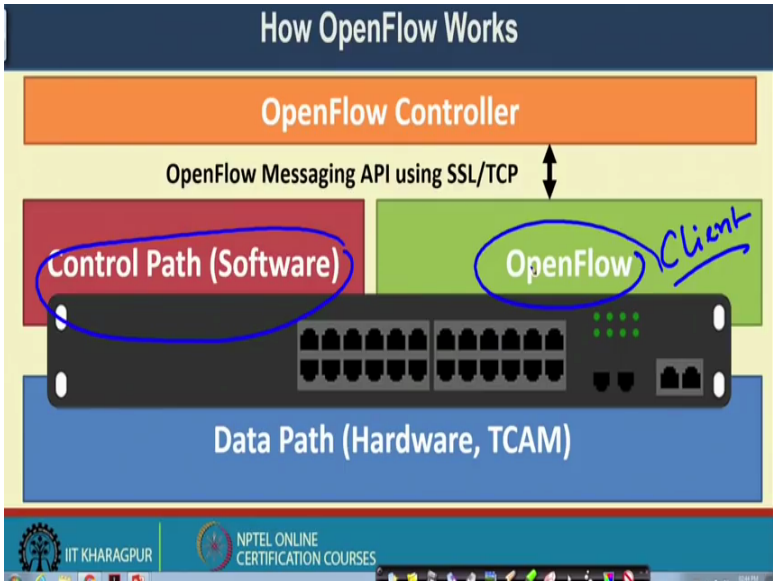
You can explore that there are this ONOS operating system, which is very popular nowadays. There is this Ryu controller, which is a lightweight controller currently many industries, they are utilizing new controller to write network programs and then other controllers like Maestro or open daylight.

(Refer Slide Time: 15:36)



So, as we are mentioning at the application layer site, you have a programming API in your preferred programming language at the control layer, you have one of the network controller ONOS, Maestro, Ryu or open daylight and at the infrastructure layer. You have this open flow in supported hardware and this southbound interface is controlled by this open interfacing with the hardware that is the open flow specification that, we are talking right now.

(Refer Slide Time: 16:11)

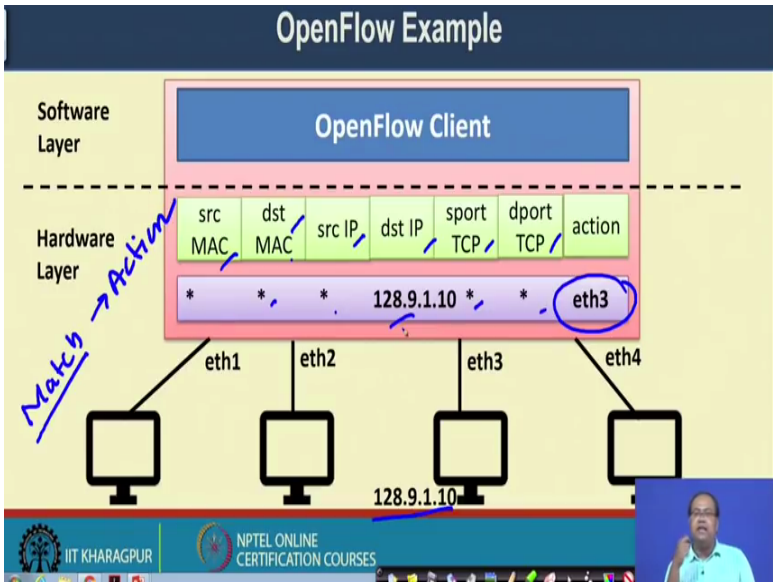


Now, let us look into that how open flow works? So, we have a switch, the inter switch as we are discussing in the last class, we have the control path, which is implemented in the software and the data path, which is implemented in the hardware or in small specific TCAM type of hardware. Now at the control path, we are having a part of the network operating system to interact the client version of the network operating system you can talk it, if it is there in this software implementation in the switch that is a kind of very minimal implementation of the control functionality is just a client version of it.

So, that you can talk with the switches and then you have this open flow protocol, open flow client, which is their inside the switch. So, you can call it as the client version of the open flow and then you have a open flow controller, which is implemented in a logical centralized machine and then this open flow messaging API, which normally uses SSL and a TCP kind of message, which talk with this open flow controller. Now the thing is that, at the software side inside switch, you have a very minimal implementation, the client version of the implementation.

So, that you can just receive a message from the controller parse the message and then configure the switch accordingly, remaining protocols the routing protocols; and all these things that, you do not need to implement inside the switch anymore.

(Refer Slide Time: 17:56)



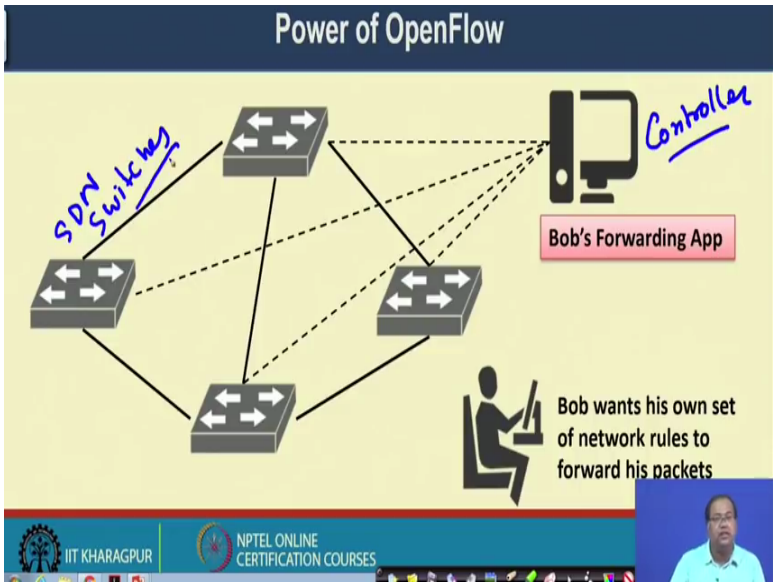
So here, is an example of open flow. So, at the switch side you have this open flow client at the hardware layer, we are maintaining a simple TCAM table.

So, this become table has multiple fields like source MAC, destination MAC, source IP, destination IP, source port, TCP source port, for destiny disappeared estimation port and the corresponding action that, you want to execute. Now a simple rule looks like this, from the hardware layer side that, your source MAC is start; that means, it is a wild card character. That means, you can accept any source MAC field, you can accept any destination MAC, you can accept any source IP your destination IP should be 128 dot 9 dot 1 dot 10, the source TCP boat and the destination TCP port can be anything and if that is the case then, your corresponding action will be forward the packet to eth 3.

So, this inter thing inter target forwarding behavior, we can write it as a match and action pair. So, we have a rule we this entire rule, that has a component of match. So, we have certain increase in the tables and then if there is a match then, you execute the corresponding action. So here, this is one entry in the TCAM hardware table. So, you whenever you are listening a packet, you extract the headers at different layers extract, the source MAC, destination MAC, source IP, destination IP, source discipline for destination TCP port, all these fields from the packet header and then make a match with this rule. So, if there is a match with a specific rule, then you execute the corresponding action.

So, the action is to forward it to eth 3. So, it eth 3 means this particular router, where you want to forward the packet. So here, the message that I want to convey to you is that any such network protocol or better to say most of the networking protocol, we can implement in the form of a match action pair, where we will see some examples of that as well.

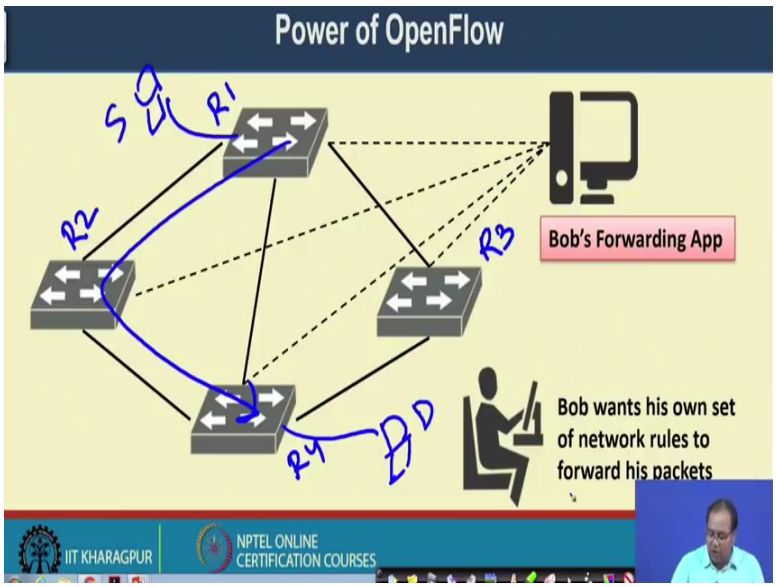
(Refer Slide Time: 20:13)



So, there is a tremendous power of this entire open flow protocol or open flow architecture. So, let us see one interesting use case of open flow. So, assume that Bob wants his own set of network rules to forward his packet.

So, we have a network controller here. So, this is the controller and these are the SDN switches, which are the dumb switches as we have mentioned.

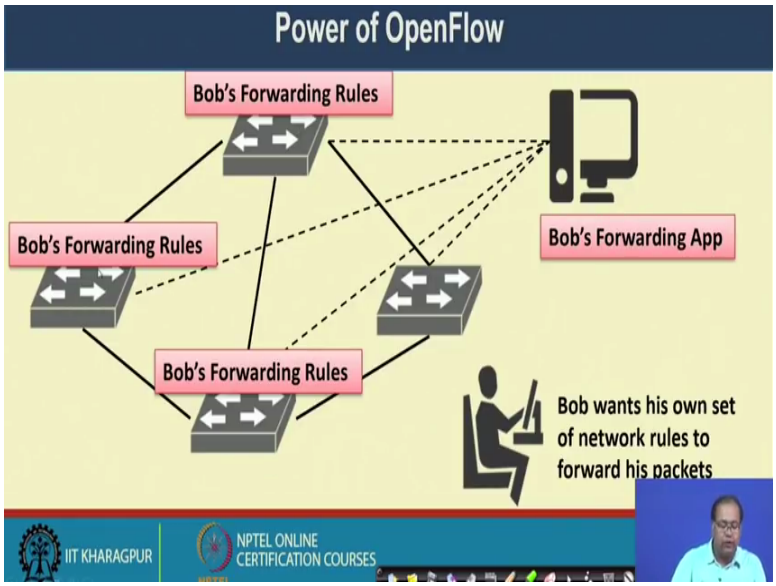
(Refer Slide Time: 20:56)



Now Bob wants his own forwarding application, say Bob wants to forward a packet from this router.

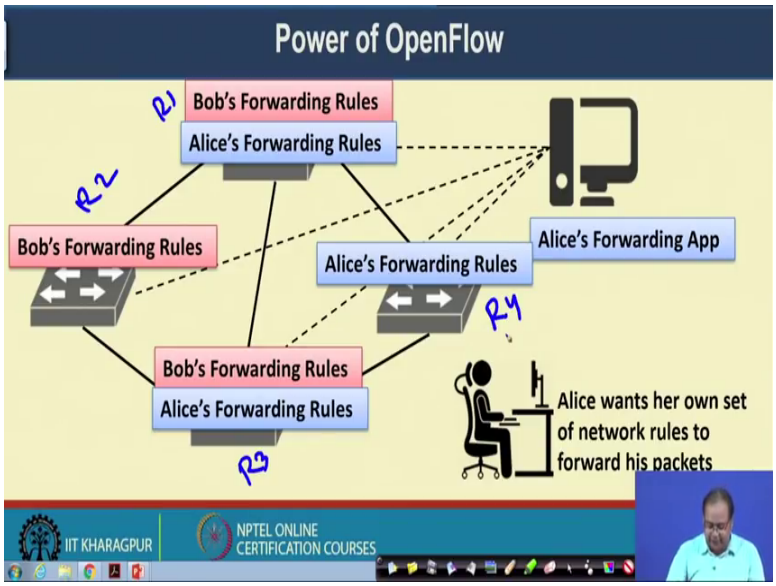
Say router 1, I am naming the routers as router 1, router 2, router 3 and router 4. Now bob wants to forward a packet from a machine, which is connected with router 1 to a machine which is connected with router 4, this is the destination. And Bob wants that the packets need to be forwarded from R 1 to R 2 to R 4.

(Refer Slide Time: 21:36)



Now, what Bob does? He basically write that descent acting in a application program inside the controller. So, the controller combines that program and after compiling the program, the controller simply D plus Bob's forwarding rule in the required hardwares. So, whenever Bob want to forward the packet these forwarding rules, which are there in the respective switches, they get executed and the packet gets forwarded.

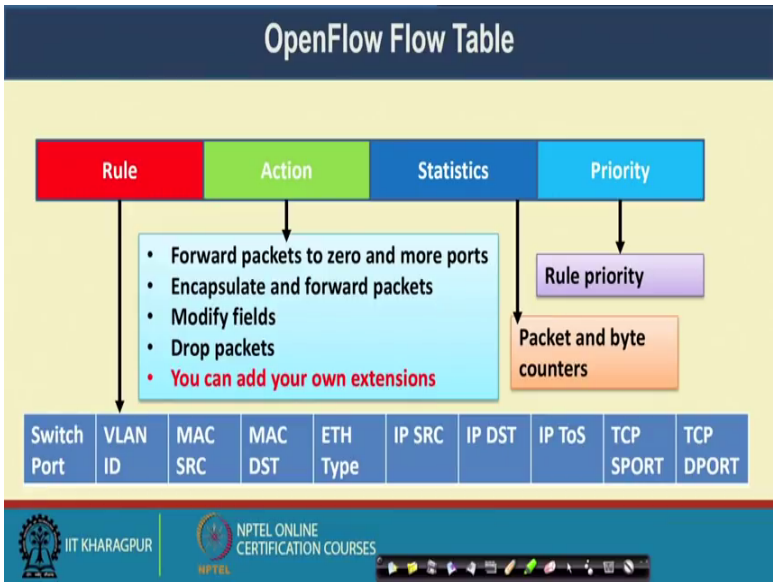
(Refer Slide Time: 21:53)



Now, when Alice wants our own set of network rules to forward a packet, Alice also program the same controller, write her own application on top of the controller and then the forwarding rules are installed in the routers on through, which Alice wants to forward the packet. Now here, you can see the interesting things that, all the routers do not need to have all the rules. So, Bob wants to use this router 1, router 2 and router 3.

So, this router or let us not use the term router, let us use the term open switch. So, Bob wants to use these 3 switches. So, the rules are installed on that 3 switches and when Alice wants to forward the packet, Alice wants the packet to be forwarded from R 1 to R 4 to R 3. So, the rules are installed in those switches.

(Refer Slide Time: 22:57)



So, if we look into the open flow flow table, the open flow flow table has 4 different component, you have the rule, the corresponding action, certain statistics about packets, the execution of a particular rule and a priority value, which is associated with a rule.

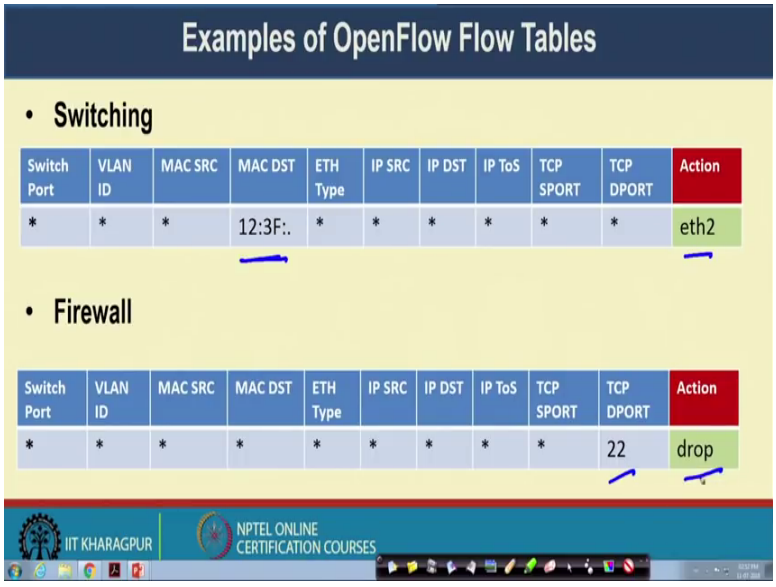
So, the idea is something that. So, you have a rule. So, the rule is nothing but a set of fields and that field basically, says that in which particular field of an IP packet or here actually, in HT and you can look into MAC, IP, TCP, all the headers. So, in your packet header which particular field to look into theoretically? You can look into any field inside the packet header. So, you can look into the packet header and our rule basically, specifies what should be or what is you are interested value for a specific field inside the packet header? Like the switch port, VLAN ID, max source, MAC destination, Ethernet type, IP source, IP destination IP type of service for quality of service, TCP source port, TCP destination port.

Now if there is a match with this rule; that means, with certain fields that, you are specifying then, there can be a set of actions and the actions can be designed by you based on your choice. So, the action can be forward the packets to 0 or (Refer Time: 24:27) ports in the switch encapsulate, the packet and then forward the packet modifies certain fields in the packet and in forward the packet drop the packets, if you want to implement the firewall rule or you can add up your own extension, whatever you can think of. The statistics fields, it have it maintains certain statistics like the packet counter,

the byte counter, number of packets that have been matched with a particular rule and so on. So, that it becomes easier for you to get the information from the network and then there is a priority value associated, which is the priority of a corresponding rule.

So, in case of a open flow, who any where you have a set of rules, if there is a match with multiple rules then the high priority rule is executed in general.

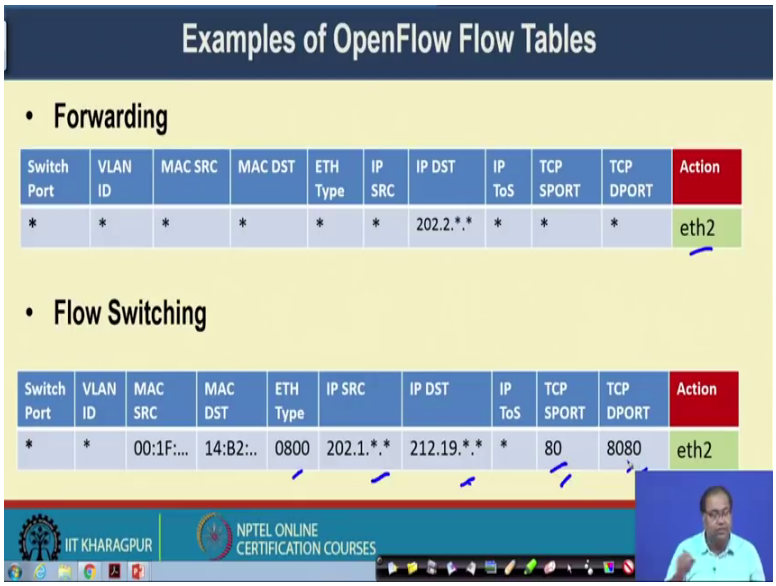
(Refer Slide Time: 25:16)



Here are certain examples of open flow tables; if you want to do a switching you have to look into the MAC destination flied, because you have to look into the MAC destination. So, you just make a match with the MAC destination, you can ignore other fields. So, we put it as a star as a wildcard character, if there is a match with this particular MAC address, you forward it to Ethernet 2, it behaves like a normal layer 2 switching mechanism. If you want to implement a firewall, you look into that TCP destination port, if TCP destination port is 22, then you drop the packet.

So, that is the corresponding firewall rule. So, you can design your own firewall rule like that. So, look into certain fields in the packet header, if there is a match with those fields of the packet header then, you drop the packet.

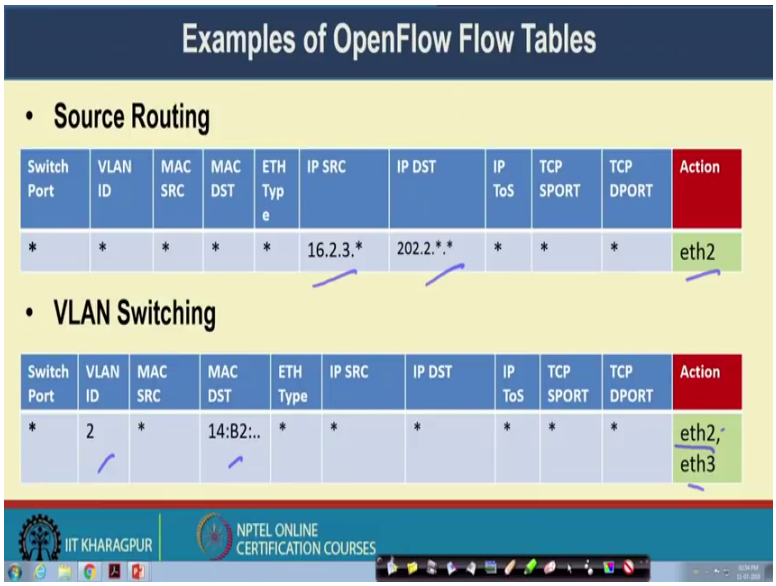
(Refer Slide Time: 26:12)



Then forwarding to forward a packet rather than looking into the MAC destination, you look into the IP destination, if your IP belongs to this (Refer Time: 26:22) 202 dot 2 dot star dot star, you forward the packet to Ethernet 2, you can make a flow switching, which is interesting. That means, this flow switching with the help of flow switching, you can make a convergence between the packet switching network and a circuit switching network. So, the idea of the circuit switching network was to use specific path for a specific flow.

Now, by looking into multiple fields in the packet header like the MAC source, MAC destination, Ethernet type, IP source, IP destination, TCP source port, TCP destination port by looking all these individual fields, you can actually uniquely identify a process to process flow, because you are also associating the TCP source port and the destination port. Now for that particular flow, you can make action that forward the packet for this particular flow to this switch. So that means, you can make flow specific forwarding or flow specific routing of the data. So, that is a huge power of STN based network.

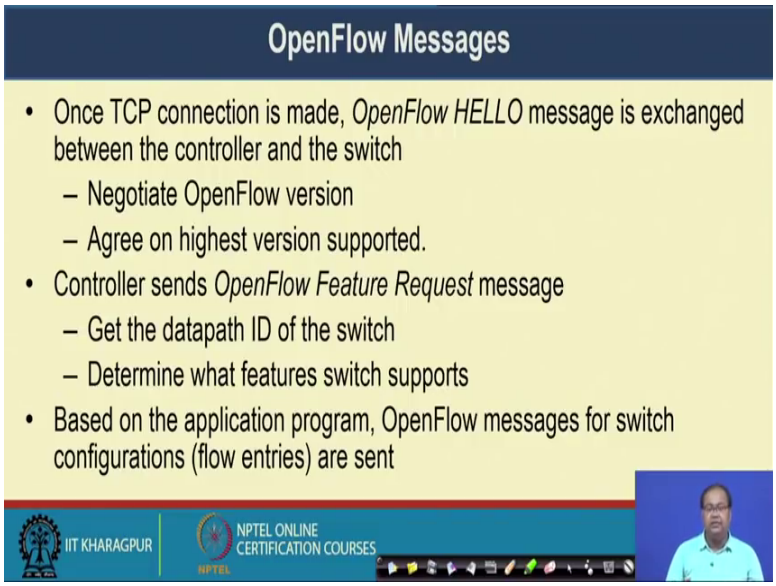
(Refer Slide Time: 27:27)



Then you can do the source routing, source routing, where if the packet is coming from a specific source and it is destined to a specific destination then, you use a specific part. So, you put source IP, the destination IP, if the packet is coming from a subnet at 16 dot 2 dot 3 dot start and if the destination is 202 dot 2 dot star dot star, the action is forward the packet to Ethernet 2. You can do the VLAN switching although, till now we have not discussed about, what is VLAN, virtual LAN? Virtual LAN is basically given a packet a set of packets. If you want to send a set of packets or a packet to a specific destination, you can forward the packets into multiple ports of the switch, which constructs virtual LAN.

So, later on we will look into the virtual LAN in details, but with the help of these open flow rules, you can specify the virtual LAN ID, the corresponding MAC destination and the action is forwarding the packet to 2 different port eth 2 and eth 3; that means, eth 2 and eth 3 are actually connected to virtual LAN 2. So, the packets will be forwarded to those interface only.

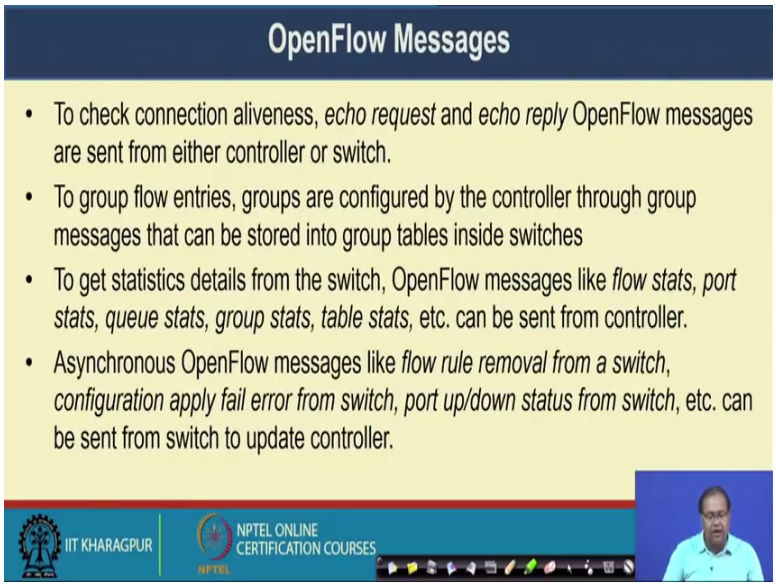
(Refer Slide Time: 28:48)



Now these are the examples of some of the open flow rules, you can design your own open flow rules and the corresponding action, the entire innovation is open. So, anyone can contribute there, let us look into the messages, which are there in general open flow the messages, which is here between the controller and corresponding switches. So, this communication, as you are mentioning they are done via TCP. So once, you have made a TCP connection between the client version of the switch and the controller, the open flow, hello messages are exchanged between the controller and the switch.

So, they negotiate the open flow version, the higher version is used and this here, certain parameters like what are the different configuration parameters you want to share? Then the controller sends a open flow featured equation message, the feature request message to get the data part idea of the switch and determine, what features are supported by the switch? Say for example, whether the suite supports QS space forwarding or not. Now based on the application program, you can say in certain open flow messages for switch configurations to update the flow to increase to modify the flow entries or to install a new flow entries.

(Refer Slide Time: 30:12)



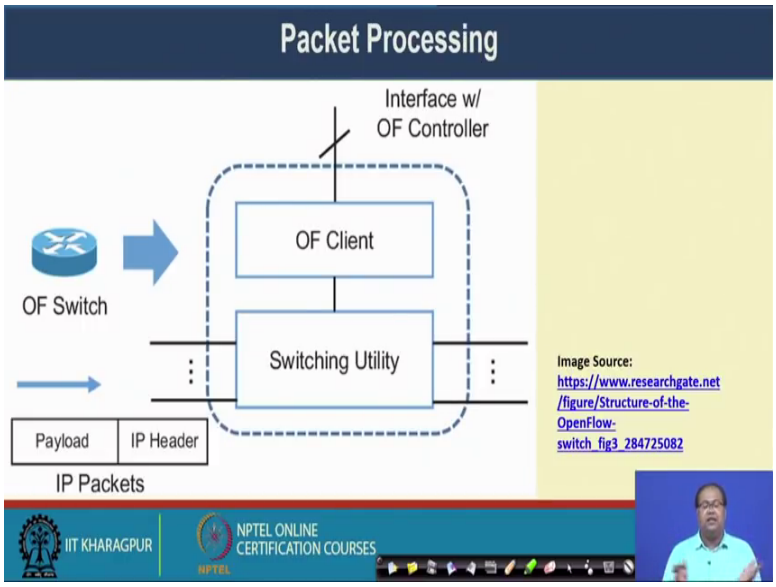
There are some other messages like to check the connection aliveness, whether the connection is alive or not open flow can send an echo request. An echo reply messages, they can be sent from the controller to switch to check the aliveness of the switch or the switch can send it to the controller to check the aliveness of the controller. Now to group the flow entries, if you want to group multiple flow entries together, these groups are configured by the controller to this group configuration messages that can be stored into group tables inside switch. So, open flow has the power that, you can combine multiple rules all together and create a group of rules.

So, the messages are there to support that group creation. To get the statistic details from the switch, you have this open flow messages like flow stats, port stats, queue stats, group, table stats, all these things that can be sent from the controller. So, that is another advantage like from the controller, you can get the statistics from the switch and you can look into the individual statistics that for a particular flow, how many packets has been transferred? And based on that you can also configure the switch for restricting the bandwidth for a particular flow; so, you have as a network administrator, you can get a tremendous power to control your network.

Then, there are certain asynchronous open flow messages like flow rule removal from a switch, configuration apply fail error from the switch, port up down status from the switch, whether a particular port is up or down etcetera, that can be sent from the switch

to update the controller. So, if a particular port from a switch is down, the switch can send a message to a controller to let it know, that this particular port is now down; so that the controller can design a fail safe mechanism for forwarding the packet.

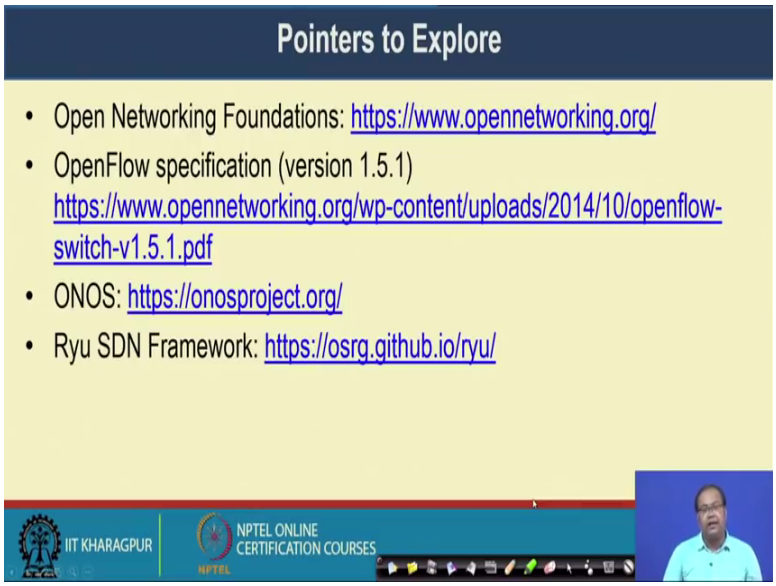
(Refer Slide Time: 32:04)



So, this is the entire a brief introduction about open flow. So now, what is happening? The entire power is in the hand of network programmer or the network administrator, the network administrator controls the entire controller write down his or her own program in the controller and this open flow helps configuring the switches.

Now, whenever you are receiving a packet, the packet has the payload and header, you look into the header based on the header filled, you make a match with the corresponding rule and then send the packet in the based on the action, which is there, which is mentioned in the corresponding open flow table. So, the rules in the open flow table inside the switch, they are implemented as a part of the TCAM hardware. So, TCAM is a programmable memory, where you can dynamically program that particular hardware to install the rules.

(Refer Slide Time: 33:16)



So, this is a very brief introduction of open flow. We will go for certain demo of open flow in the next class before going to that. So, I am just giving you certain pointers that, you can explore yourself to look more details of this entire open innovation in the networking community. This is a kind of advanced topic in network and you should learn that because, people predict that our future network is going to be SDN controlled.

So, there is a link for open networking foundation, where you can find out the different standards, different agendas which are there under the open networking foundation, the open flow specification the current version is stable version is 1.5.1. You can look into the different messages, their messages type, their functionality, all these thing. The ONOS, ONOS is an popular network controller, you can look into the ONOS details, it is open source thing again, you can just install it in a single machine and start using it, there is another SDN controller called Ryu, I suggest you to look into the Ryu controller as well.

So, these are all open source tools, you are free to download them, free to use them. So, explore them that is all about the course today.

Thank you all for attending the course. Happy learning.