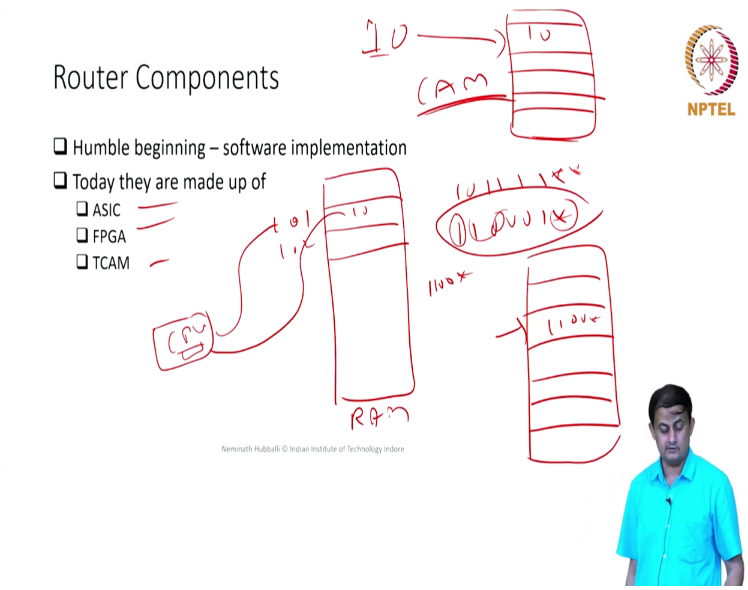
Advanced Computer Networks
Professor. Dr Neminath Hubballi
Department of Computer Science Engineering
Indian Institute of Technology, Indore

Lecture 2
An Introduction to High Performance Switching and Routing - Part 2

(Refer Slide Time: 00:19)



What is there in a typical router inside; how do you implement these lookup operations and other things like quality of service policy and another thing that I was talking about. So, as usual, the routers also had a humble beginning, where things were implemented as a software solution.

We write the code and program to do this lookup operation basically, this is a string matching and searching operation, and as days passed by and the network speed typically turn increasing, the software solution didn't scale up. So, it turned into hardware, and there are several hardware mechanisms that came into the picture.

So one is something called the ASIC; ASIC stands for an application-specific integrated circuit, which means that I design the circuit or the hardware specifically to do the lookup operation, so it does only that thing, and nothing else and we built such very customized hardware, then we can scale up the operations that we bring, and hardware is anyway is faster compared to the software.

And on top of that, if you build very custom hardware, not the generic hardware like the one we have in our typical computers, we only need to do that routing and forwarding decision. So, you bring the scale up the operation. And typically, people also talked about FPGA-based implementations. FPGA is another kind of hardware technology that is in use. And the third one is something called the TCAM, TCAM stands for ternary content addressable memory. So, there is a difference between the CAM and the TCAM, and the typical RAM that we have. So, let us quickly understand what this is.

The RAM that we have in our computer is called random access memory. So, what it typically is : there is an address to every location; let us say 101 is this location and, 102 is this location and so forth, then we want to access something from this memory, Typically the CPU gives the address, can you go to this location 101 and get the content that might be probably here: if like for example let us say I have ten here, so, ten will be brought to a register inside this CPU, every time I want to make this access, some content from memory, I keep giving reference of the address, so go to this address find this, go to this what is there, you get me that and go to this location and write this content, so, from the CPU register the content would be transferred to the corresponding memory location.

But unlike this, what CAM does is, so I am not going to tell you the address by giving the address, but what I give you is I want this content. So, can you search for this content, whether in one of your locations or more than one locations? If this is the content, I am looking for ten and can you find out, you quickly tell me whether ten is there inside your memory? It would be anywhere. So, whether it is in the first location, second location, third location or last location we do not care. As long as it is there so we are fine, so we just want to find out whether this is there or not there and if it is there, where it is there, which location it is, that is the question and CAM, the content addressable memory can precisely do this operation. So, you give me the content; I am going to search inside my memory locations, whether that content is available or not. If it is available, I am going to tell you this is available and where it is available, if not available, then I will tell you, this is not available.

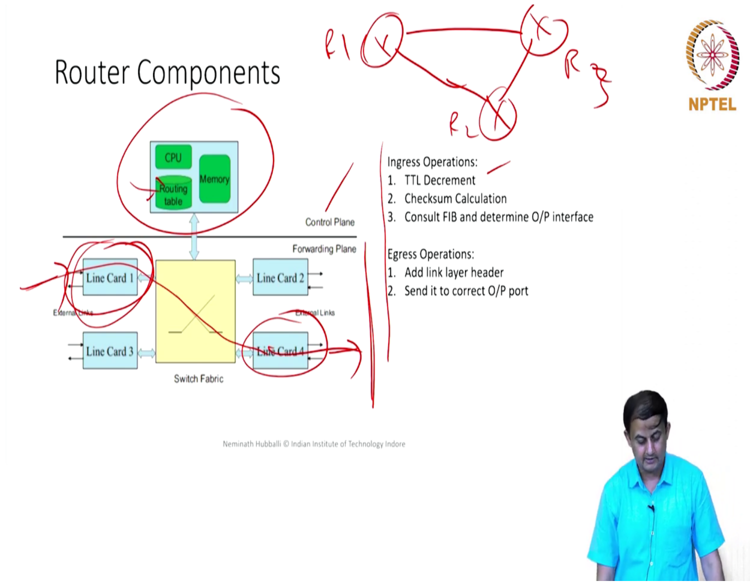
So, think of this case, I got an IP address, and I want to do the lookup operation. Using this memory, I can tell that router, I have got a packet with so and so IP address, so, can you quickly find out whether for this IP address, where you have a routing entry inside this, if yes, what is this location and then what is the port number on which I need to send this. So, that is the kind of operation, we are looking at.

So, CAM perfectly fit this case, but what TCAM says is, typically we talked about the routing entries being in the prefix format, the entries are not exactly in the binary format of 101111 something like this, but there are also entries like *, so, 101111* or something like that and it might have 110001* and TCAM allows you to do the search operation or the lookup operation using such kind of prefixes.

So, every bit can be either 1 or 0, but it can also be a * I can give a third component which is the * also as an entry. So, I am looking for 1100*, can you find out? So, it can quickly go and find out 1100* is here. So, what is it, next on which port number I need to forward this, it can quickly come back and tell you so that you can make the forwarding decision.

So, such memory is content addressable, but it is not binary content addressable, but it can also take these * into account. So, it can search keeping the * or the prefixes, using the prefix directly. Using a prefix system, it can search, that is what, in our context it means. So, such TCAM memory is also available with the routers nowadays using that, you can actually bring scale-up operations to the routers, typically to the border routers and the core routers.

(Refer Slide Time: 06:57)



So, that is what typically the router contains. If you look at a little more detail about what exactly a router does, as I said the typical operations of the router are divided into two parts, one is the control plane operation, and the second one is the data plane operation which is also called the forwarding plane operation.

In the control plane operation, you have got the routing table, this routing table is being updated. I got a memory and CPU and typically every router keeps exchanging the information i.e., the routing updates to its peer. So, R1 sends its routing table to R2 and R2 sends to R3 and R3 sends to R1, R1 sends to R3 and they keep exchanging this information, so that will tell you whether they have connectivity to this route, that route, this prefix, that prefix and based on that they keep updating the routing table. So, that is what is happening in this portion of the router component.

And we can use typical software algorithms, the normal CPU that you find in computers, and all that, so this is perfectly fine there but on the second hand, this forwarding plane operation as I said needs to be done in real-time. So, I need to have something more. So, the things in the forwarding plane are: one is we have something called the line cards, these line cards are typically from the cable that you connect to that router.

In addition to that, there are some more components, we will see what is there inside that line card in a little more detail. But there are a bunch of line cards where the cables come and terminate. And let us say an Ethernet cable or an optical fiber cable terminates, the interface where that cable terminates, and then in addition to that there are some more things. And then the second component that you have is something called switch fabric which tells you if I receive a packet on this line card and I make taking the reference to the routing table that is loaded into this forwarding plane operation, so I decide I need to go to this exit line and then forward this packet, the packet comes here and then goes to this interface and then exit the router.

So, the switch fabrics allow you to connect from one interface to another interface. From an incoming interface to the outgoing interface, it establishes a link. If I want to go to line card number 10, how do I go there from the input line card number 1, and if I want to go to input line card number 3 and go to the output line card number 4, how to actually go there? So, such interconnection is actually given by these fabrics.

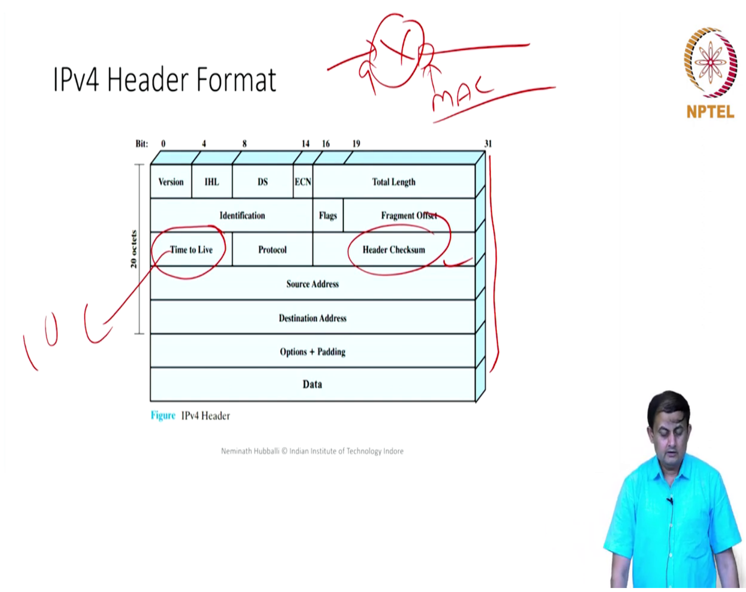
So, the typical operations that are done inside, between the packet arrival of an interface card and the time the exit happens, there are a couple of things; one is after every hop, you need to do the TTL decrement, i.e., you reduce the TTL value by 1. So, that is one thing. And if you reduce the TTL value, then you need to re-compute the checksum value. At each hop of the router, you need to re-compute the checksum. So, that is done typically at the input line card here.

And once you decrement the TTL value, re-compute the checksum, you extract the bits corresponding to the destination IP address, do the lookup operation, and then quickly decide, I need to go to output port number 4, and this is the corresponding line card I want to use, and then you push the packet to that particular port number. And at the output link, when the packet actually comes to this line card, you need to do something more.

What is that? Add the link layer header; that means on an input link you have a link layer header added that is stripped off at the input line card. And from this route onwards to the next hop, you can add all together a different link layer header. So, can we think of this case, the first hop I am a wireless link using WiFi or a typical GSM network and I am in the packet that is a wireless app I have a different link layer protocol from this hop: my computer to the nearest router. And then from there onwards to the next hop, you are entering the core of the network, you might have a different link layer protocol that is working.

So, typically link layer protocols are meant for one-hop operation. So, I have one link layer protocol and typically the corresponding link layer header and the first hop. And I have a different connectivity in the second hop and a different link-layer header in the second hop. So, what is the next hop link, the link between the corresponding router and the next hop. Depending on that, I am going to add the corresponding link layer header and you send the packet on that particular link. So, this is typically what happens in the forwarding plane or the FIB operation.

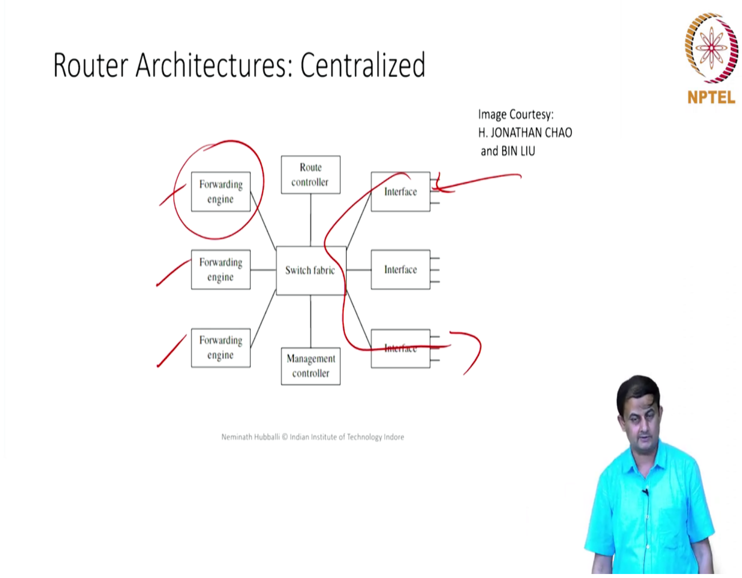
(Refer Slide Time: 12:35)



Here is the IP version 4 Header. So, this is the thing I was referring to. You have got the time to live value (TTL). This we also call hop count; if it is set to ten when a packet arrives at a router, the router makes it as nine and then sends it to the next hop. Here is where the line card corresponding to this output link is, where the MAC header or the link layer header corresponding to the next hop is actually added.

And since the TTL value changes, the header checksum is calculated on the entire header from here to including the error. So, the checksum value will also change, you need to keep updating that also at every hop.

(Refer Slide Time: 13:27)



So, now, the next question that we want to ask is how do you actually build the router, there is a control plane operation, and there is a forwarding plane operation or the data plane operation we want to call, and these are two different things. So, is it that routing table I take and keep it in one place, and then every time I receive a packet on any port, I go to that routing table and then do the forwarding decisions? Or there is a multiple number of the instances of the routing table itself, and then I can make an independent decision.

So, if I am a line card, if I receive a packet, then I can quickly consult, if I have a local copy and then making forwarding decisions quickly. So, depending upon whether a centralized repository of the routing table is kept inside a particular router or there are multiple instances, we categorize such router's architectures as either centralized or distributed.

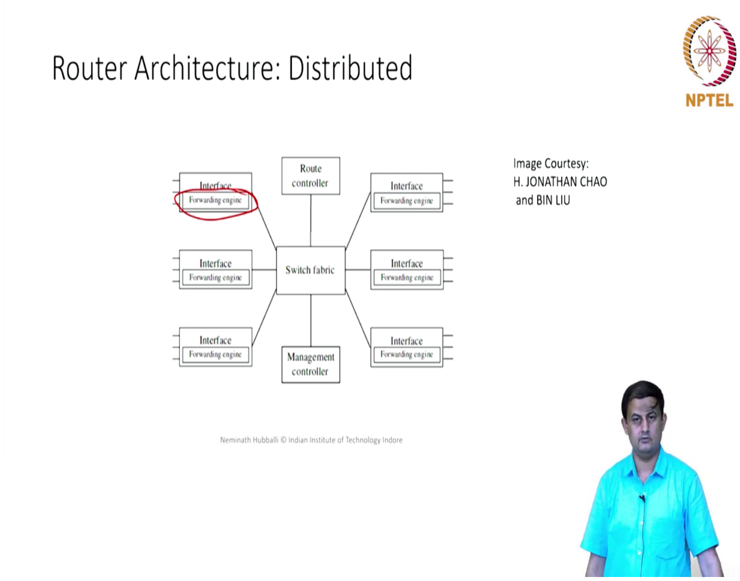
So, in the centralized routing infrastructure, what we have is: some instances, not necessarily one; here in this diagram, there are three copies of the routing table. So, there are three instances that are going to do the lookup operation, and we got some number of the interfaces. So, you receive a packet here, and you access one of the available forwarding engines at this time, whichever is free; you access that, consult that routing table and do the lookup operation, decide where to send the packet, and then through this fabric, you actually route the packet and send it.

So, if you use such an architecture, then it is called centralized architecture. And then, along with the switch fabric, the switch fabric is connecting the two different interfaces, you got something called a route controller and also a management controller; the route controller is meant for constructing that routing table and dealing with the routing algorithms in the control plane operation. And the management controller is meant to manage the configuration of the router if I want to change the metric.

So, for example, so typically routing algorithms, will we use the hop count and other things as the metric? Say If I want to update that metric or change the routing algorithm, I need to configure that router. The interface between the user and that particular router is exposed to this something called the management controller.

So, I will summarize again. The routing controller is meant to deal with the routing algorithms. Management control is to do with the configuration of the router. Switch fabric interconnects the different interfaces, and the forwarding engine typically does the lookup operation and tells on which output interface this particular packet needs to be forwarded.

(Refer Slide Time: 16:30)

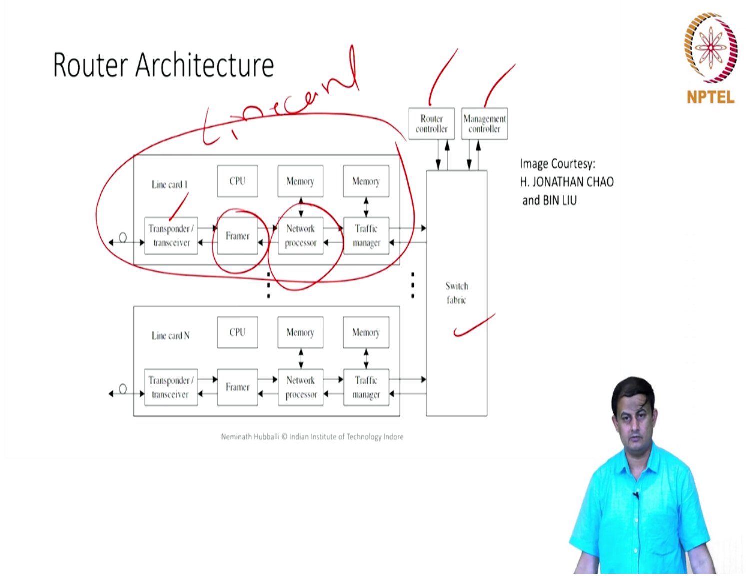


So, in the distributed architecture, what we have is we still have the route controller, we still have the management controller, we still have the switch fabric, but on every interface, we have a loaded forwarding engine. So, there is a routing table sitting here with a processor it has got everything. So it can do the lookup operation independently. In the previous case, there are only a handful number of instances, and everyone is sharing these handful number of the instances' routing table and then doing the forwarding decisions.

But unlike here, I have an independent routing table sitting in each of the interfaces and they are being consulted every time a packet is received on that interface. So, you can think that by replicating these instances, I am going to actually, already if I, let us say have 100 interfaces and then nearly 10 forwarding engines. So, these 100 interfaces need to share only these 10 forwarding engines, but here if you have distributions, every interface has got routing table. So I can bring scale-up operations. So I do not have to wait for that forwarding engine to become free to do the lookup operations.

So, typically, nowadays, whatever the routers that you see, we follow this architecture of the distributed structure where every interface has got the instance of the routing table loaded along with the necessary network processor to do the lookup operations.

(Refer Slide Time: 18:10)



So, the typical line card that I was talking about will look something like this: switch fabric is here, the router controller and management controller are here, and one line card will have these many numbers of the elements. So, it has got something called a transponder or transceiver. So, this is an electrical or optical device, which actually receives the signal on a link and then it knows how to interpret that. So, for example, if you have an optical fabric cable, the data is transmitted in the form of light.

There is a signal that is going inside that fiber optic cable. You need to interpret that, this is when 0 is coming, this is when 1 is coming, and a sequence of zeros and ones coming, and the job of this transceiver or a transponder is to understand the incoming signal and then decide whether this is 0-bit or 1-bit, bits are coming serially inside that link. And if you are a sender, then you either send a 0 or 1 depending upon the content of the packet.

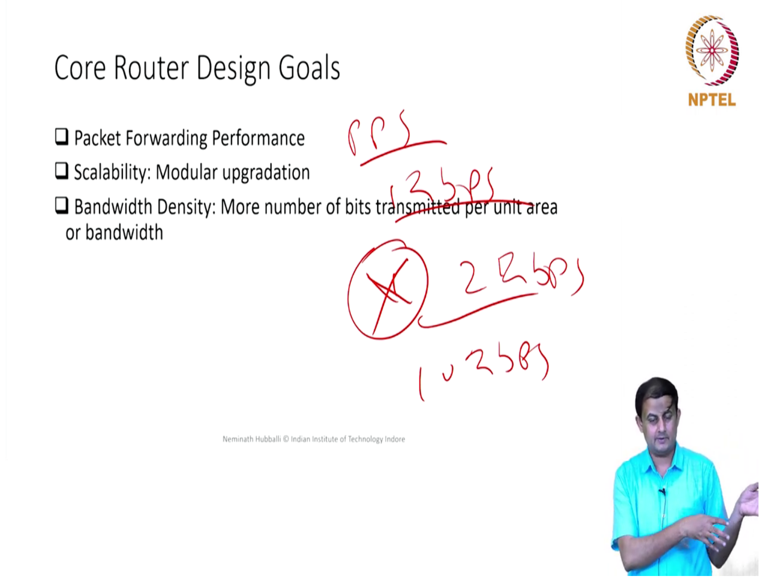
So, you also have something called the framer; this is where layer two, the link layer operation is coming that is also typically implemented as part of the line card. So, if you receive a packet, you need to understand the header structure of the link layer and then appropriately interpret how big this link layer header is and where the next layer IP header is starting, and from which offset I need to pick up the bits so that I look at the destination IP address and make the decisions and so forth.

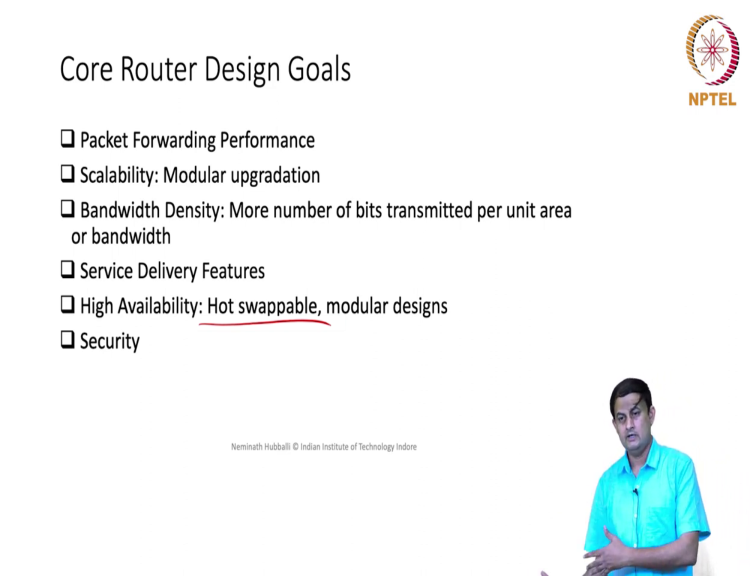
To understand that you have got a framer, on the other end, if you are a sender, you are forwarding a packet, you need to add that corresponding link layer header that is done by this framer. And there is a processor that is sitting that we call the network processor, which does this lookup operation; because you want to do the lookup operation, there is some computation involved; you are typically searching for something.

So, that is done by this network processor, and it has got an associated memory. And you also have something called traffic manager, the example that I was talking about along with the forwarding decisions, routers also need to do the bookkeeping operations like who is sending at the maximum rate, who is sending at the reduced rate, whose traffic I need to forward, whose traffic I need to block. So, all these operations are controlled and done by these traffic managers.

These are all the typical components that you see on a typical line card including a processor, that is the network processor, framer, and traffic manager. So, that is all about the routing architecture and the components. So, these all put together will bring the scale of the operation that we are looking at.

(Refer Slide Time: 21:22)





So, with that background in mind, routers need to do a lookup operation and some other kind of bookkeeping work. And these routers need to do these operations at a massive scale with millions of packets per second. And if I want to design such a router or a switch, what are the typical design goals which a router needs?

So, there is a bunch of them; let us take a look at them. So, one way to evaluate the performance of the router is how many numbers of packets this router is able to forward in real-time. So, packet per second might be one metric with which you can evaluate, or alternatively find the designer; I need to aim for a maximum number of the packet per second processing capability.

The more such capability we have, the better it is. So I can do the operation quite quickly, which is required in the backbone and the border routers.

And the second thing that you want to aim for or would like to have in the router is modular upgradation. This means that we know that things are going to be added, more number of the users are going to be connected, more number of the prefixes are going to be added. So, the size of the routing table is increasing, the number of people is increasing, the number of hosts is increasing. So, if I am the network service provider, how do I design my router?

So, one thing is ten years down the line; I anticipate that my routing table entries or size of the routing table is going to be size x. So, looking at that ten years down the line, I am going to build hardware or a router that is capable of handling the traffic, and the scale-up operations required ten years down the line, and that will require a cost.

So, ten years down the line, what is going to happen? Now, only I need to do the capital investment. I need to bring the router which is capable. On the other hand, if you have a modular design, currently, my traffic is looking like this: let us say I am receiving 1 Gbps traffic on my router, I am going to build a router that is capable of handling 1 Gbps and two years down the line, this might to go up to 2 Gbps.

So, I am going to add new hardware, new component to the existing router which can handle 2 Gbps of the capacity, and ten years down the line, if I am aiming for 10 Gbps, so I keep adding. Every year I am going to add some components to the existing router. And I extend the processing capacity, lookup operations, everything that I want to do with the router by adding the components not one go but in batches. So, that is going to help me I do not have to buy the hardware right now itself but by looking at the traffic growth permit in next ten years.

So, that is actually anyway if I buy the hardware for looking at the traffic growth in the next ten years that is going to be underutilized. And you also end up putting a lot of money into that one. So, by doing this scalable, modular upgradation capability into the router, I am going to actually help, I can bring the components as and run and then keep on increasing the capacity, I am going to add line cards to that router new line cards as and when new people join, new connectivity is going to be there. And I can also add new CPU , new memory, whatever I want ,everything is going to be modular.

And the third thing that I look at in the router is bandwidth density, meaning how many number of the bits that I can handle parallely in one go. Bits in the sense, packets is one terminology, packets are entirely made up of the bits, the more number of the bits processing capability that I have in my router, the better it is. The size is compact, but in the same size, I am going to handle more number of the bandwidth, maximum bandwidth that is better for router.

And in addition to forwarding or lookup operation is the primary job of the router that is what I said. But in addition to that, the routers also need to do a lot of other things like the policing or quality of service and all that, the more number of search features the router is able to handle on the existing capacity, the better it is. So, that is called the service delivery features.

So, anything other than the lookup operation, whatever you want to do, if the router is supporting that, the better it is. And then, these routers, now think of the situation I am talking about a router connecting the internet service provider networks of several nations, maybe one link from the India, another link from the USA are coming and terminating at a point and the router which actually interconnect these two links need to be highly available meaning that 24/7 365 days it has to run.

The moment these routers are down even for minute, then it will disconnect the internet service provider for that much time. So, I need to design these routers for very high availability in a fault tolerant fashion, so that I can do the operation, but I can guarantee that this router is going to function 24/7 365 days. And in order to do that, the cloud has typical routers that are available in the market or having a capability something called as hot swappable.

Hot swappable something like this, I do not have to shut down the router to do an upgradation. Typically let us say, if I have a computer and I want to insert a new RAM card and my mother board, how do you do that, you shut down the computer, open the mother board and then on a slot, you fix that RAM and then restart the system, when they start to function it start working. But that I cannot afford to do on any typical backbone or the border routers on the scale-up operation. So, I do not want to put the router, even rebooting is also not possible.

So, in that case without setting down the router itself, I can bring the new hardware, plug in there, and it will start working. So, that is the kind of features that I am looking at, looking at the very high availability. And then of course, the routers, they do use routing tables, they do have the memory and the CPU to do the processing. And these operations need to be secured. Meaning that, a false router entry should not come there, somebody should not get hold off, they are able to manipulate the routing table, influence routing decision that should not happen.

So these are some of the design goals. These are some of the objectives with which you design router. Now the question is how do we actually achieve this in practice, if I want to do a quick route lookup, what kind of the data structure that we used to do the route lookup and what kind of the hardware do we use to do the route lookup? If I want to do the quality of service, how do we actually how do we implement that in the software and in the hardware and what kind of algorithms are being used that is the content for the subsequent lectures for this module. We will see some of those examples in the subsequent lectures. So, I will stop it here for this session. Thank you.