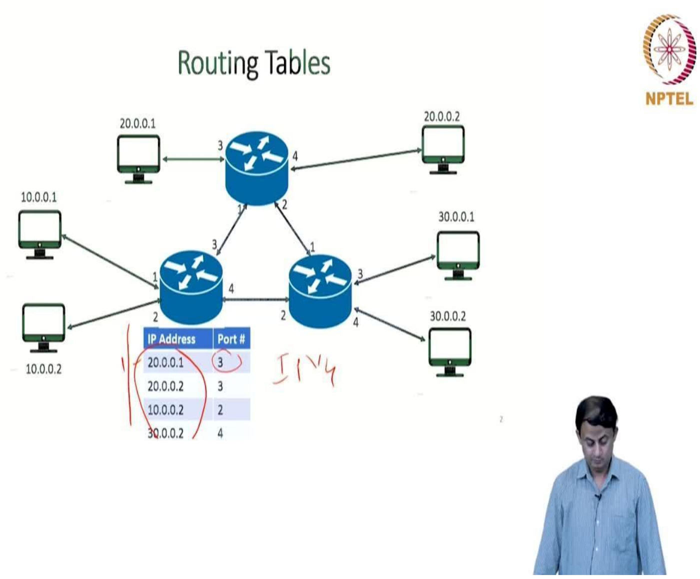
Advanced Computer Networks
Instructor. Dr. Neminath Hubballi
Department of Computer Science Engineering
Indian Institute of Technology, Indore

Lecture 3
IP Table Lookup – Part 1

Welcome back. In today's lecture, we will discuss a topic called IP table lookup. IP table lookup is one of the primary operations the routers in the network need to do. So, let us see how the routers actually do this kind of take lookup operation, what are the complexities involved in the lookup operation and the details of that.

(Refer Slide Time: 00:44)

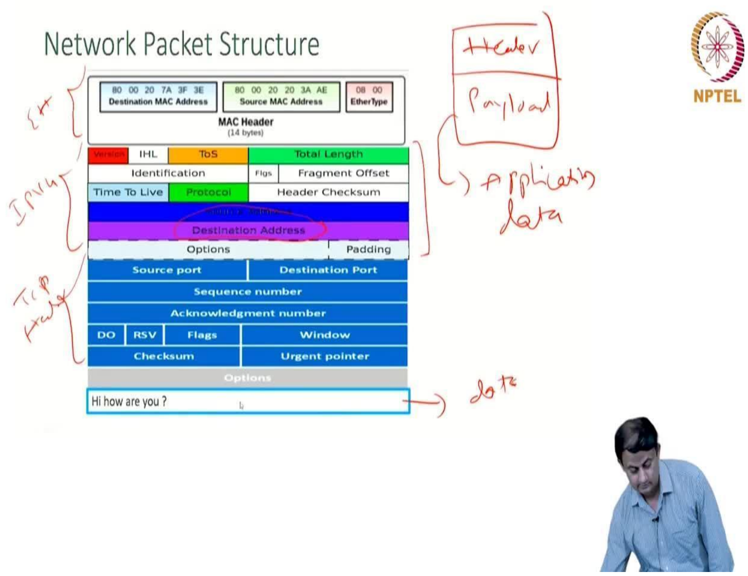


So, we saw last time the diagram similar to this. So, what you see here are a bunch of computers and three routers, and we see that the IP address table or the routing table in the router looks something like this. So, this is a simplified version of the routing table what you have is a bunch of IP addresses, so this is the IP address number 1; if you receive a packet with this destination IP address, you need to send it to this port number that is the meaning of that.

Similarly, if you receive a packet with this destination IP address, you need to send it to the same port number and so forth. So, every router in the network actually has such routing tables, the routing algorithms actually periodically exchange information about their neighborhood and connectivity and then come up with such routing table and the forwarding operation is basically to consult this routing table and then do the forwarding decision. So, where to forward this?

Now, the question is, what is happening, when I see that you need to look into this table and then forward it? What exactly is meant by that? what exactly happens inside the router to actually do this forwarding operation? So, you can see here on the left-hand side that every IP address that is over here, or the complete 32-bit IP address, this is the IP version 4 address. So I have got a bunch of them, and then the next column shows the port number on which that packet needs to be sent.

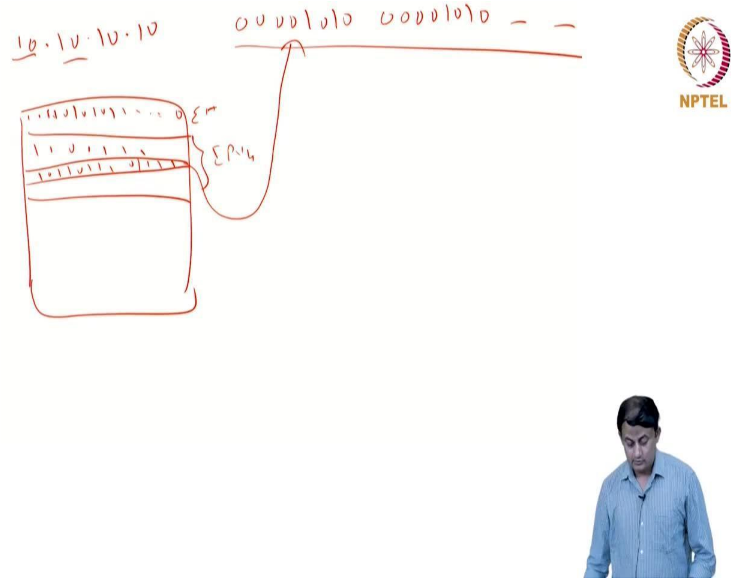
(Refer Slide Time: 02:41)



So, when I receive a packet, typically it has got two parts, the first part in the packet you got the header and then the second component is called the payload. So, every packet has got some portion which is the header part and the second portion is the payload. Payload is the application data. So, on the left-hand side, you see this header there are different headers this is the portion where the actual content is going that is the payload or the data that is carried inside this packet. From here to here is the Ethernet header and from here to here is the IP version 4 header and from here to here is the TCP header.

So, in the routing, this is the forwarding decision that the routers need to do, they operate using this particular header and particularly the destination IP address, which is here, you pick up the bits corresponding to this destination IP address precisely 32 bits and then compare that with what is there inside the routing table. So, you have got these routing entries and so, these are all 32 bits right now and the destination IP address is also 32 bits and then you compare, it looks something like this.

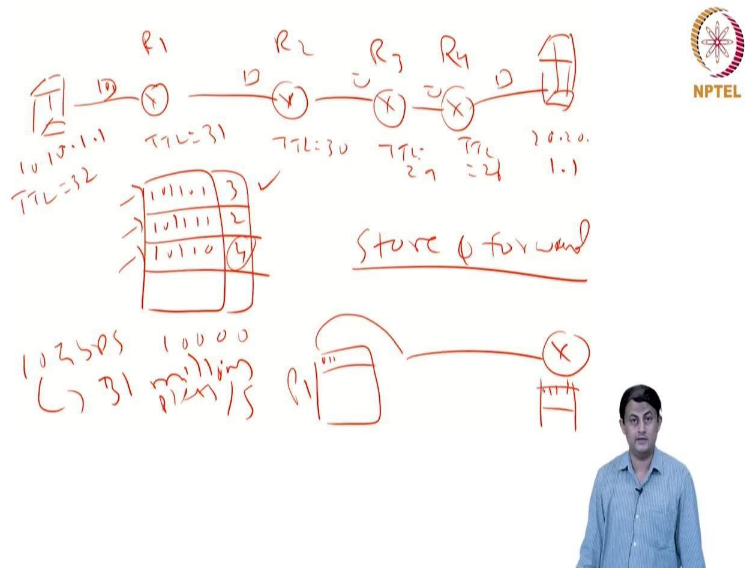
(Refer Slide Time: 04:37)



So, if my IP address entry is 10.10.10.10 and you convert this into binary, 00001010 is the first component this one and then another 00001010 for the second component similarly, you got the third one and the fourth one. And when you receive a packet at the router, there is a sequence of bits. So, this is something like this. So, this is corresponding to the Ethernet header, and then there are some bits corresponding to the IP version 4 header, and out of that, somewhere here are the bits corresponding to the destination IP address.

So, what it means is, I am going to take these 32 bits from the IP version 4 header and then compare it with the entries that I have in the routing table. So, every entry looks something like this in the binary format, I do a string matching by picking the bits from the IP version 4 header to that what is there in the routing table. So, that is precisely what is the lookup operation. So, for every packet that arrives, you need to do this lookup operation.

(Refer Slide Time: 06:17)



Another thing to note is when I say that the packet is actually going from one computer to another computer, let us say my computer is here somewhere in India, and this has got the IP address 10.10.1.1, and it is going to another computer which is having the IP address let us say 20.20.1.1 and this packet is routed through a bunch of intermediate routers, let us call them as R1, R2, R3, and R4 and a TTL value is initially assigned. So, let us say this is 32, every router decrements the TTL value by one, and here when the packet arrives, which is exiting from this computer over here, let us call that packet P1. So, the TTL value here becomes 31, the same packet when it arrives here, the TTL value becomes 30, and when it arrives here, the TTL value becomes 29, and when it arrives here, the TTL value becomes 28, the same and finally, with the value of 28, the packet is actually delivered to the destination.

This is one part of the operation, and at each router, there is a routing table where you need to do that consultation process. So, every routing table has got an entry like this, and we have some binary sequence and then the corresponding port number, then the second binary sequence and running port number, and the third binary sequence like this and then the port number. When any packet say P1 arrives here, you pick up the destination IP address, consult here, compare with this one, compare with this one, compare with this one and wherever you find the match, you send it to that particular port number.

So, this operation is happening at every router, in R1 and R2, R3, and R4. So, one operation is you are decrementing the TTL value by one and then sending the packet, and the second operation that the router needs to do is to pick up the bits corresponding to the destination IP address and then compare it with the entries in the forwarding table that are there.

So, now, it is obvious that in the last class, we said that the network speed is increasing, meaning now there are networks with links that are capable of transmitting at several Gbps rates if that is the rate at which the packets are coming to the router, and for every packet, I need to make this lookup operation. What is that it takes?

So, one obvious thing is, as the number of entries in the routing table grows, meaning so let us say, I have got 100 entries in the routing table. In the worst case, if I am doing a linear search on this table to find out a route or find a port number on which I need to forward the incoming packet, there are a hundred 32-bit comparisons that I need to do.

And on the other hand, let us say there are 10,000 entries inside this table, in the worst case I need to do 10,000 32-bit comparisons. So, as the number of entries inside the routing table increases, you have a lot of work to do. So, how do the entries in the routing table grow? When new people join the network, there are new IP addresses which are seen as many as large number of the people join. So, routing tables are going to grow in size.

So, with that, on the one hand, the link speeds are increasing and on the second hand, the number of entries in the routing table is also growing. So, if that is the case, this is actually going to put a lot of challenges on the router. So, I need to do the lookup, do a large number of comparisons and at the same time, I need to do this operation as quickly as possible i.e., in as less time as possible, because the link is actually pushing a lot of packets onto the router.

And the second thing to embrace is when I say this, forwarding, and the TTL is decrementing, this all operates with the concept called the store and forward, what it simply means is, the bits on the link are arriving sequentially, meaning if this is my particular packet, and if I start transmitting from the left-hand side, this bit is going to arrive first and then if this is the router, and there is a buffer here, and this bit comes and sit here occupies space here, and the next bit arrives in the next sequence that will come and sit in the next position and the third bit will come and it will sit here something like this, sequentially bits are arriving on the link and they are getting stored inside the buffer precisely, what it means here it is called a queue.

Inside a queue, the bits corresponding to that packet are coming and sitting, and till the entire set of the bits corresponding to this particular packet arrives here, you cannot actually do the processing. So, in the worst case, at least the header portion whatever is there, that portion all the bits corresponding to that should come, the 32 bits corresponding to the destination IP address should come, then only you will be able to do the comparison.

So, at every hop, the storage operation is happening, you receive a packet, store it, then pick up the bits from this buffer, and then go and consult this routing table and then decide where to forward. So, if I am my destined source is here, and the destination is let us say 20 hops away; 20 times you need to store the same packet and then consult the routing table 20 times and then finally deliver the packet to the destination. That is actually a lot of work to do. So, 20 different routers on the way actually need to do this lookup operation.

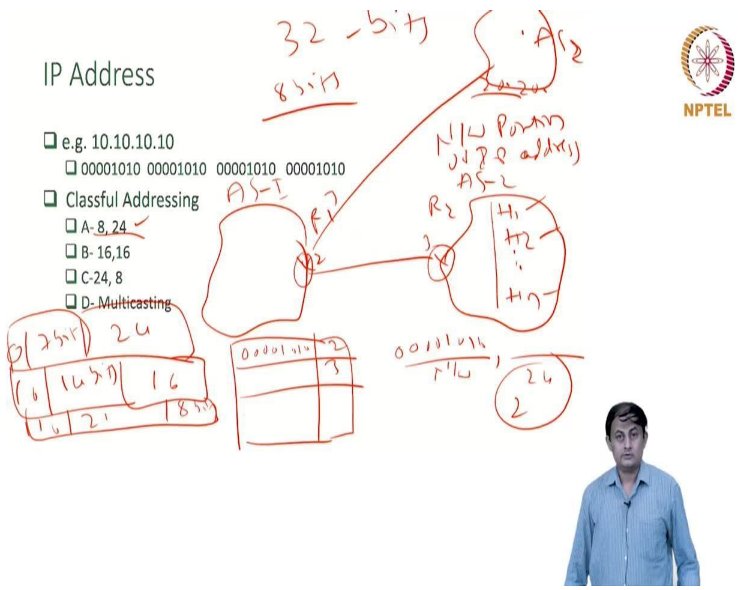
So, if these routers are not capable of handling these lookup operations or if they are not able to do this forwarding decision quite quickly, then the end-to-end delay would increase. If let us say, every router takes one millisecond to do that, and if I am crossing 20 hops to reach the destination, then 20 milliseconds is the time just taken to do the lookup operation, and over and above, you got the processing time, you got the forwarding transmission time and other things they will add up. So, you will experience a lot of delay during the transmission. So, that actually the effect.

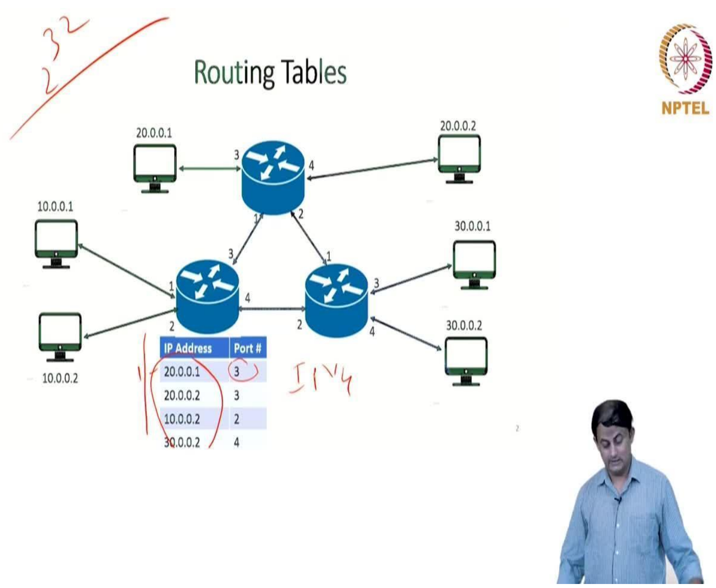
So, if this is what is happening inside, now the question is, in order to fully exploit the link speeds that are available in a network, what is that I need to do inside this router to do this lookup operation quite faster? So, as soon as the packet comes, I need to do this process quickly and then send it.

To give just an estimation of what is the kind of processing that we are looking at. If you have a link that is operating at 10 Gbps, that will bring approximately 31 million packets per second. So, what it means is, in order to forward all the packets or fully utilize the 10 Gbps capacity or the bandwidth that I have, the router needs to do 31 million lookup operations per second, that is the scale of the operation that we are looking at. So, this is actually quite a complex or very challenging operation to do.

So, we will try to understand how exactly this is done. What kind of algorithms or mechanisms I can use inside the router to cope with this kind of peak high-speed lookup operation?

(Refer Slide Time: 15:38)





Before we actually jump into the specific algorithms, let us look back at the history a little bit and then derive some motivation for how exactly these routing table entries are organized, I see that, at least in this example, I showed there are 32-bit entries. So, it is not necessary that in this routing table, every entry need to be of 32 bits size, there is some optimization.

If every entry is of 32-bit size, then every router on the internet needs to have 2^{32} entries. So, that is actually quite a huge routing table. And if you want to do a comparison, if you are doing a linear search, then it means that 2^{32} 32-bit comparisons you need to do. That purely is not scalable. So, we need to do something other than that.

So, what the network organizers or the engineers talk about is, let us organize these IP addresses into some kind of hierarchy and name them into some classes. So, Class A address, Class B address, and C address. What it means is if I am having a network here, and if there are a bunch of computers inside, let us call them H1, H2, and so on Hn, and all these computers in H1 to Hn share a common prefix and that I denote as the network portion of the IP address.

So, let us say the network portion if it is a Class A address, so what it means is a Class A address reserves the 8 bits which are common for the network and the remaining 24 bits for the host, it is 8 and 24. The first 8 bits are common for all these n numbers of the host, and the remaining 24 bits are actually varying. So, what this means is if my sequence is something like this 0001010, and this is the network version, and the remaining 24 bits from here to here are the whole IP address.

So theoretically, you can have 2^{24} numbers of different hosts inside this network. So, what it means for the routing and the lookup operation is: Think of this as a case, this is the autonomous system number 2, and this is the autonomous system number 1, and there is a router at the border of this autonomous system number 1, and there is a router which is at the border of the autonomous system number 2 and these two are connected with the link.

And given that all these hosts inside the autonomous system number 2 share a common prefix of the IP address, which looks something like this, assigned to typically all the packets which are going to whether they are going to H1 or H2 or to H3 or to so on Hn, they have to take the same route.

If this is port number 2 and this is port number 3 for this router. This is R1, and this is R2, and what the router R1 can do is: I do not care really what the 24 bits host component is, let me build a routing entry with only the network portion of the IP address, and I will say that every time you receive these first 8 bits are 100001010 you need to forward that to the port number 2 or the interface number 2.

So, by doing this, what actually we did is: the routing table entries are supposed to be 32 bits and then we found it is good enough to store only the 8 bits. So, this will help by doing this, by combining large sections of IP addresses, and only one entry corresponding to that is actually stored inside the routing table.

And similarly, you can think of this so if there is a third network or autonomous system, and this is using the IP addresses, let us say 20 dot series, I can create an entry only one entry for the entire series and say that you need to forward the packet on the interface number 3, so whatever the common prefix is there if 20 is the prefix you come and enter here.

So, a bunch of hosts in this network, in this network for all of them, we actually stored only one routing entry in this router R1. So this will reduce the number of entries in the routing table. So, that will bring a scale of the operation, meaning that as the number of entries shrinks, you will be able to do the decision quite quickly because you do not have to search all the routing table entries if there are only 100, then only 100 comparisons are required.

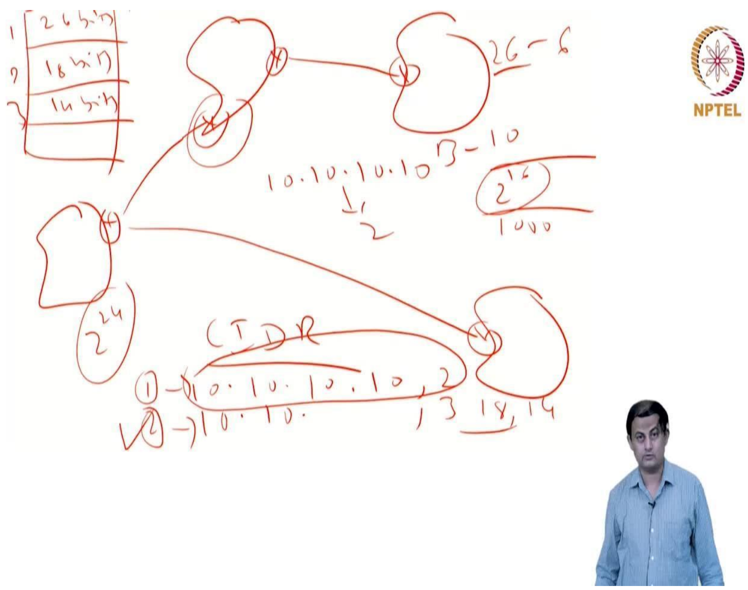
And moreover, what it says is instead of doing a 32-bit comparison, now, I am using a Class A address it is good enough to store only 8 bits; only 8 bits comparisons are required. So, by doing this, to some extent, you can accelerate or speed up the operation of the lookup operation. So, that is due to the Class A address. The Class A address is identified with a leading 0, and then the next seven bits are varying, and then the 24 bits. This is the seven bits together, this 0 together this is 8 bits, and the 24 bits are for the host component.

Similarly, if there are networks that do not require 2^{24} number of IP addresses, you can go for the class B address which reserves 16 bits for the host component and then 10 is the one that is fixed and then 14 bits are reserved for the network component. So, 10 is fixed and 14 bits are actually varying. So, they can take from all 0's to all 1's, these 14 bits.

Similarly, the class C address is identified by 110 and this reserves 24 bits meaning including this 110 and then 21 bits for the network portion and then the last 8 bits for the host. So, if you use Class C address, then the maximum number of hosts that you can have is 2^8 and then the maximum number of networks or distinct autonomous systems that you can have is 2^{21}.

And if you use Class B then you use 14 bits, i.e., 2^{14} different number of the networks and 2^{16} different distinct hosts within each of those networks. So, the question is, if I use Class A, then the number of the routing table entries probably that I require is 2^{8}. If I use Class B address, then the number of routing table entries in the router is 2^{16}. And if I use Class C address, then the number of routing table entries would be 2^{24}, 2^{24} is also quite a big number.

(Refer Slide Time: 24:01)



Assuming I have a network where every autonomous system on the internet is actually using a Class C address. What it theoretically means is the routers which are here, every router, which is in the border connecting two autonomous systems needs to have 2^{24} number of the entries that is actually huge. So, this is 2^{24} entries here, 2^{24} entries here, and so forth. So again, as the number of hops between the source and destination increases, the searching time at each output is going to increase.

Now, one aspect is I want to put this number of the entries in the routing table within the limit and that is why you want to do some optimization that is one aspect of the optimization that I want to bring. And the second thing is these classful addresses are actually kind of wasting the IP addresses that are available for utilization.

So, let us say that this autonomous system is using the Class B address, which is 10. And if it does not have 2^{16} number of hosts, let us say it has got only 1000 hosts, then the remaining sequence of the IP addresses which are part of this series would not be utilized, they go waste. So, in order to avoid this situation, what the network designers thought is, let us not fix the IP addresses: the range, this is the network, and this is the host portion, and let me make it variable.

So, that is the concept of something called classless inter-domain routing in short form this is called the CIDR. So, every autonomous system or every network has to decide, depending upon the number of hosts that are going to be part of that network, what portion of the 32-bit IP addresses I am going to reserve for the host component and what portion of the IP address I am going to reserve for the network portion.

So, what it simply means is, instead of saying that this is a Class B address and I require 16 bits for the network, what it can say is, I have some x number of the host, I require 26 bits for the network and the remaining 6 bits for the host component that is the division. So, 2^6 is the number of the host that is there, so somebody else can say that I am going to have 18 bits for the network and the remaining 14 bits for the host component, and then you can have 2^{14} number of hosts inside that network.

So, if the number of the bits for the network portion is not fixed, what it means is, the routing table entries are also not going to be of the same size. So, if everybody was using the 8-bit IP network IP address, then the routing entry would also be 8 bits. But what we are saying is now one can have a 26-bit prefix, and another can have 18 bit net ID.

So, what it means is if I am the router, then in my forwarding table, entry number 1 can have 26 bits and entry number 2 can have 18 bits and entry number 3 can have 14 bits, and something like this. So, within the routing table, different entries can have a different number of bits stored inside the table, every entry is of different sizes which is possible.

So, in that case, my comparison, what I want to do are going to be even more complex. So, you do not have any alignment meaning that I know I am not saying that you do only the 8-bit comparison, you do only the 16-bit comparison, you do only the 24-bit comparison and so forth. So, one is 18 bits another one is 26 bits another one is 14 bits, I need to do a variable number of the bit comparisons within the routing table.

And moreover, it is possible for the networks to have one entry which is 32-bit which is more specific to a particular host, and another entry for the same network with the prefix stored inside header or what it means is, it says so if let us say 10.10.10.10 is the IP address, this is the exact 32-bit IP address. This is entry number 1 in the routing table and entry number 2 might say that you forward it to port number 2 and anything that is starting with 10.10 and I do not care what the 16 bits are, you forward it to port number 3 or port number 2 or whatever it is. So, this is possible.

Then when I am doing the search operation, when I am doing the string matching, comparing the 32 bits. So, if I find a match this one might go with this one or I should go with that one, one is saying that you should forward it to port number 2 and the second one is saying is it forward it to port number 3, whether to follow the second routing table entry or the first routing table entry. That is the question.

And to the answer to this is, the routers are supposed to do the longest prefix matching, meaning that if I find an entry in the routing table, which is the maximum that is matching with respect to what I have as the destination IP address, you follow that. So, what it means is, in this case, if you receive a packet, which is supposed to go to the destination 10.10.10.10. And then, although this is a valid match, the longest match is this entry, you forward this particular packet to port number 2.

And what it means is a consequence of this is: I might have a large number of entries given one destination IP address, and the 10.10.10.10 can match to multiple number of the entries in the routing table. So, meaning that one is saying exact 32 bits, you forward it to port number 2. And the second is saying, if the IP address is starting with 10.10, then you forward it to port number 3, and the third one might come back and say the IP address is starting with 10, then you forward it to port number 4, something like this. So, it is not good enough to do the comparison, and you find a match and then do the forwarding.

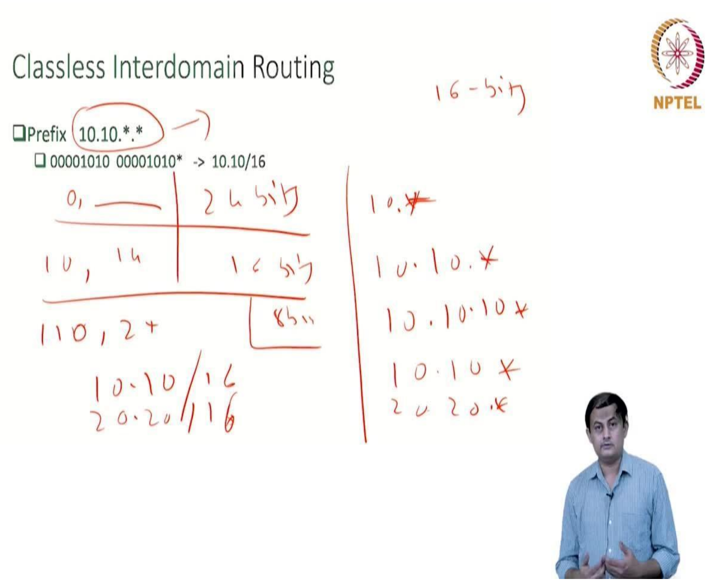
So, I should keep doing the matching and find the best possible match for this particular destination IP address. And using that, I am going to do the forwarding decision, so that is the thing. So, although let us say my number of entries in the routing table is n and I start comparing the destination IP address, I start comparing entry numbers 1 and 2, and so forth.

If, let us say 10th entry matched, it might be a fraction of that, it might be in this format 10.10. And then you forward it to 3, although this is a valid match, you are not supposed to stop the search operation or the comparison operation there itself. I do not know, there might be subsequent entries, which are more specific, and it might match more number of the bits than just the 10.10.

So, in that case, what is the possibility in the worst case that can happen, so you need to compare every packet with every n number of the entries inside the routing table. So, that is actually going to put a load on the router. So, every time let us say, I receive the packets, and most of the pockets are able to find the route in the first 10 entries, then although I might have 1000 entries subsequently if I am not searching them 99 percent of the time, then that is a substantial reduction in the searching work that the routers are doing.

But that is not happening here. So, you need to continue to do searching till you find the best match. So, I do not know where is the best match. So, that is going to be complicated, so you need to search all of them.

(Refer Slide Time: 33:21)



So, that brings us to the concept of something called prefix-based routing. So in effect, what we said is, if I am saying that my network address is 16 bits, the routing table entries are 16 bits. So, if it starts with 10.10. And then what I can do is I can denote that or the entries inside the routing table using this prefix notation 10.10.* is actually a prefix notation.

So, all the classful addresses, for that matter, when I say that class C address is starting with 0 and then the next seven bits, and then the 24 bits for the host portion. This I can represent in a prefix format like this: 10.* is the prefix, so, that is the address. And then, if I am using the Class B address, then the first bits are 10 and then the next 14 bits, and then the 16 bits. So, that might bring us to the prefix format of 10.10.*.

And if I am using the class C address, then this is 110 followed by 21 bits and then the remaining 8 bits. So, the prefix might look like 10.10.10.*. So, all these appear in the form of the prefix, and so is the CIDR entry something that is written as 10.10/16 would correspond to a prefix which is 10.10.* and something which is written in the format of 20.20/16 would mean 20.20.*.

So, whether you use the classful addressing scheme or the classless addressing scheme, depending upon the network portion that is available, I can denote that in the form of the prefix. What it means is when I am doing the search operation, I am actually doing the search operation on the prefix notation, and the routing entries are in the prefix format.

And the comparison is also done by taking the, what it means is, when I say /16, you ignore the next 16 bits, pick up the first 16 bits of the incoming destination IP address, and then you compare it with the entries inside the table. So, that is how the comparisons are done.