Communication Networks

Prof. Goutam Das
G. S. Sanyal School of Telecommunication
Indian Institute of Technology, Kharagpur

Module - 12
Network and Transport Layer
Lecture - 58
DHCP and Routing

So far we have seen how two layers can cooperate with each other and then how the address resolution can be facilitated. So, now, what we will try to see, we will talk about we have already started talking about that there is means as the time progresses, there is a huge crunch of or giving or distributing the IP addresses. Because too many devices are coming up with the advent of IOT and the internet of everything probably a huge number of IP addresses as required.

Already there is a solution that IPv6 with a 128-bit address, but that is a long-term solution of course, but currently, as we have talked about the network should be interoperable, and the network should be gradually moving toward that final destination or final modification so that is one thing. The second thing is it must coexist with the previous one being still there and with that we should be able to resolve some of the issues.

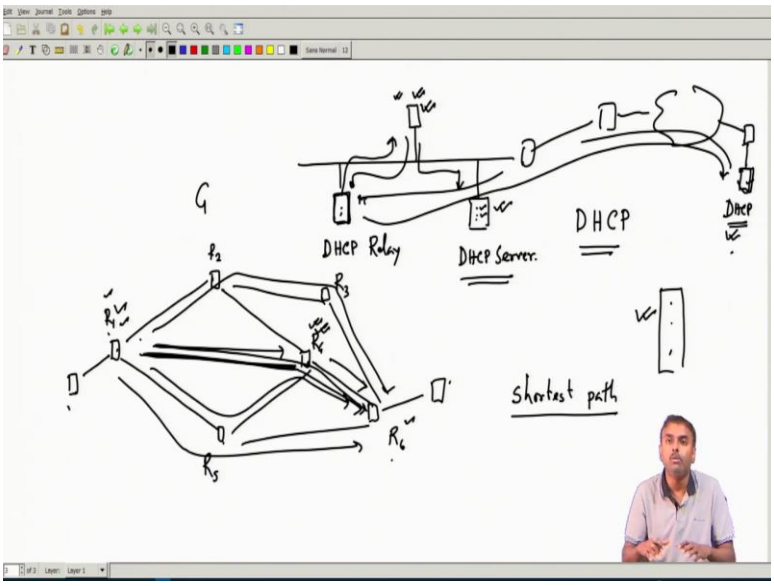
So, that is something which we will be talking about, which means meanwhile that people have thought about resolving this issue of IP address crunch. So, what happens, it is first let us talk about how I provide IP addresses to users. Suppose user is a first-time routing, at that point how can I give him the IP address, if he is in the network, he needs to have a unique ID otherwise his identification is not unique. So, nobody can forward the packet to him and he will jeopardize the operation of the networking.

So, everybody who is connected to the network is connected to LAN and then through LAN, through router probably in the internet he must have a unique ID, otherwise, we have a big problem ok. So, if that is the case one way is that whoever is routing that machine, so actually first goes hardcore the IP address over there, he has to keep a repository for which IP address I am distributing where this is a very cumbersome process, it is a manual process, it might be erroneous.

The person might misplace what he is putting where he might forget he might put accidentally a duplicate address. So, that will create a huge amount of problems. Whomever he wants to forward the packet if there is duplicity the router will not understand, he will learn that ok this particular IP address is connected to some other port, and he will forward it accidentally to others.

So, proper networking will not be happening. So, if that is the case can we automate this process? Initially, people started thinking about that. So, what is the process of automation? So, automation might be something like this IP address repository I can keep it somewhere ok? So, we can actually distribute that. So, we can call that an address repository or a server that keeps that we will give the name later on after the concept is fully developed, but right now we can talk about that repository.

(Refer Slide Time: 03:48)



So, somebody in the LAN, suppose inside a LAN somebody is keeping all these IP addresses, a list of IP addresses that can be distributed among the machines. Whenever the machines are coming up. So, they are first-time routing. So, immediately what can happen? They might, now it is the reverse thing which will be happening ok? So, basically, what will happen because the machine has its own network card so it has its MAC ID, the IP address might not be there; so the MAC ID will have. So now, what he can do? Just the reverse of ARP.

So, he might, with that MAC ID he might say this is my MAC ID, you can broadcast because he knows that the central repository which has the list of IP addresses that have to be allocated, is already there in the land. If it is there in the land then he can just broadcast it, ok, hoping that this guy will respond back ok. So, he will give the MAC ID and he will say I do not know my IP address. So, can anybody just broadcast this message, can anybody help me what should be my IP address?

So, once he broadcasts that will also go over here because it is a broadcast domain. So, the entire land all the packets will be transmitted, he will see that he will see which IP addresses are already being allocated and what are free. Among those free one of them he will be taking, which is why it is called dynamic host configuration protocol. So, that is why this is also called a DHCP server. So, the DHCP server repository keeps all these addresses and also keeps a flag on whether this ad address has been allocated or not allocated.

If allocated he means puts an entry that this is already allocated this cannot be further allocated, otherwise, there will be duplication. If some of them are not allocated he picks one of them randomly, that is why it is called dynamic host configuration. So, for the host he is configuring the IP address, so that is why it is called dynamic host configuration protocol.

So, he picked that ID, and then he replied back, with his MAC ID he had already supplied his MAC ID he told me that this was my MAC ID I did not know my IP, and then he said ok. In layer two he replies back with his MAC ID that this is your ID and that should be your IP address.

So, immediately he takes; that means, layer 2 packet, he learns the IP address, he puts the IP address inside his particular OS and network card ok. So, immediately he will know his IP address. So, this is how he discovers his own IP addresses. What is the advantage of this? We will talk about why we have been talking about this IP address crunch, and how this helps we will talk about that later.

So, this might happen. Now, the problem is that every LAN must have a DHCP server, this might not be happening your entire institute might have a DHCP server or it might be somewhere else. Somewhere your internet service provider might have that DHCP server ok somewhere in a wan, in some remote server. Now, every station can only broadcast, they cannot broadcast packets everywhere because if everybody starts doing that the entire network will be flooded by those broadcasting messages and the entire network will be choked. So, generally, this broadcasting is not allowed beyond your LAN ok?

So, you can only broadcast messages within the LAN. So, that is why every LAN will have at least one machine configured as this DHCP proxy ok. So, almost similar to our ARP proxy. So, DHCP proxy what it will do? It will actually take that message and it knows the IP address of the DHCP server, wherever it is it might go beyond the router, beyond the even gateway to the internet somewhere else. Whatever it is this DHCP server has an IP address, that IP address he knows.

So, he knows the address of the DHCP server. So, he becomes a bridge between this guy's connectivity towards the DHCP server. So, because this guy does not even know the IP address he can only generate that layer to broadcast telling me this is my MAC address I do not know my IP address somebody can help me.

So, this proxy will take that ok or DHCP relay we can call it, this relay will take that he will then put this MAC ID and tell ok corresponding to this MAC ID can you supply a particular IP address, he will forward that packet. Now, that packet is going from him only he knows his IP address is MAC address, he knows this IP address. So, immediately he can transmit this packet.

So, that packet like earlier we have discussed. So, that packet will go through the entire network to the DHCP server, The DHCP server will give that IP address to him then he actually replies back saying that ok. MAC ID I have resolved it should have this IP address and he will give that IP address and whatever IP address he takes DHCP server already will be aware of that, because he has only configured this IP address. So, he will tick that IP address which has been taken by this particular host.

So, every time a host comes up, he will be taking these IP addresses from the DHCP server and this will continue. If this happens why this is advantageous? This is advantageous because it is like this; this DHCP server what he can do? He can be like the network service provider. So, the network service provider will resolve some of the IP addresses, but he will subscribe to multiple more users compared to as many addresses he has, and he will be hoping that not all the users simultaneously will be using his service.

So, with that hope and generally in networking that happens, in the peak hour also you will see only few users among all the users will be utilizing it ok. If that is the case it's kind of statistical multiplexing of addresses. So, you have a limited number of addresses, and now you have so many users, but you know that all users will not be active simultaneously.

So, what you do for the time being you give him a temporary address ok. So, you give him an address, whatever time he is using it you keep him with that address and then whenever he releases it, then immediately you take that address as if it is available and give it to someone like that.

Now, next time you come up and again try to connect you might not be getting the same address. You might be getting another free address which is currently available with the DHCP server. So, this is what the DHCP server does and that is why it is also called dynamic host configuration protocol, and with this, you might be also means helping yourself towards reducing the unnecessary blockage of IP addresses.

So, basically, you take a huge pool of IP addresses, whichever users are active you randomly allocate them all these addresses and then whenever somebody is releasing it you take that, you declare that as if that is free, and the next user asks for it you randomly again allocate those addresses.

So, this is how DHCP at least at the beginning helps IP to reduce this or to combat this address crunch of IPv port. So, basically you can you can serve a large pool of users without giving them all unique IDs, you can actually take a large number of again smaller than that large number of users some number of IP addresses which are unique and then you randomly allocate them to all these guys. So, what will happen while they are communicating, they have a certain IP address for the entire session they will have that IP address at that session which means all the packages will be delivered correctly to them.

Whenever the session expires he actually means he goes away from the network he is no longer utilizing then the DHCP server will take that IP address and give it to somebody else and whenever you require he will be giving some other things at that session your IP address will be that only. So, the IP address of the host keeps on changing over time, but it does not hamper his communication with the network, at the time of his communication he will have a unique IP address because that is what the DHCP server makes sure.

So, this will be happening. So, that is how DHCP at least solves one part of IP address crunch. We will also see NAT, which is again a cross-layer thing, with the transport layer that also resolves some of the issues of IP address crunch, by locally using some locally unique ID, but not globally unique ID. So, that can be reused in multiple localities which is something we will see, whenever you are doing local routing that is possible, and global routing is also possible with the help of the TCP layer the above layer.

But before that, we should also be starting to discuss how these routing table entries are being learned, how the routers in a distributed fashion know if I have to send this packet to somebody which should be the next router, what should be my next stop which port I should connect to. How this is being done, is something that will be discussed. This is a distributed protocol generally routers exchange messages, and control messages among themselves.

So, if I have a complicated network, let us say I have a graph G, where multiple routers are connected among themselves in a complicated networking fashion or network connectivity ok. So, these are all routers, let us say router 1, router 2, router 3, router 4, router 5, router 6 ok. So, what I need to know is if a particular host will be connected if, he has an IP address another host is connected he has another IP address. So, from here to here if I go what should be my strategy which router I should choose from hop to hop? So, he must know where I should forward.

So, if he is choosing this path, then he will be giving it to him then he must know where I should forward next otherwise, they might take a longer route, unnecessarily the packet will be delayed and unnecessarily all the routes will be congested because you are all packets are going where longer route, its staying in the network for a longer duration within that time more number of packets will be coming. So, the entire network will get more and more congested.

So, what is this particular thing that routers do among themselves in a distributed fashion and they resolve this what should be the next hop and then what should be the next to next hop and all so on? Every router decides what should be the next hop for a particular source-destination pair of a packet. So, this is something we will be trying to understand.

So, basically, this boils down to a problem that is well known as the shortest path finding ok, shortest path computation, or finding the shortest path between two nodes. So, from one router to another router what is the shortest path? Now, shortest might be an ambiguous definition, what is shortest is it distance-wise shortest or is it delay-wise shortest? So, it by means as you define different things accordingly the shortest path definition also will be changing.

So, if I talk about physical distance, then I have to find out what is the smallest physical distance because in a very complicated network, there are multiple paths to reach from R 1 to let us say R 6 I want to reach, it can be this one, it can be this one, it can be this one, it can be this one. So, there are multiple paths through which I can reach a particular router. Now, the distance probably will be if you inspect if you see, of course, I have to provide the distance of each of the links we will see that this is the shortest distance.

Therefore, if I decide that this is the shortest distance. So, whenever there is a request from router 1 to router 6, always I should go via router 4; that means, router 1 should know that among router 2, router 4, and router 5 which are his immediate next neighbor, I must choose router 4. Because I am trying to reach router 6 and router 4 also must know that among my neighbors, like router 5, router 2, and router 6, he must choose router 6.

So, this is the shortest path because this is the shortest distance. Sometimes it might be just a delay, I might try to see what is the delay happening in the overall these things. Locally I might be able to decide what should be the shortest path ok? Sometimes it is not the local shortest next hop, as you can see I have to go from R 1 to R 6 the shortest path includes the longest next neighbor. So, remember this is not as simple as we are thinking about, it should be seen end to end and then only I can come up with the shortest path strategy.

So, in a very complicated mesh architecture with multiple routers maybe 1000s of routers or maybe even more, some millions of routers with a very complicated network with a lot of edges and this many nodes, how do I construct from a particular node to another node what should be the shortest path. So, there are multiple you probably are aware of; there multiple such shortest path algorithms which are there. One of the famous ones is Dijkstra ok. So, we will try to see that Dijkstra one.

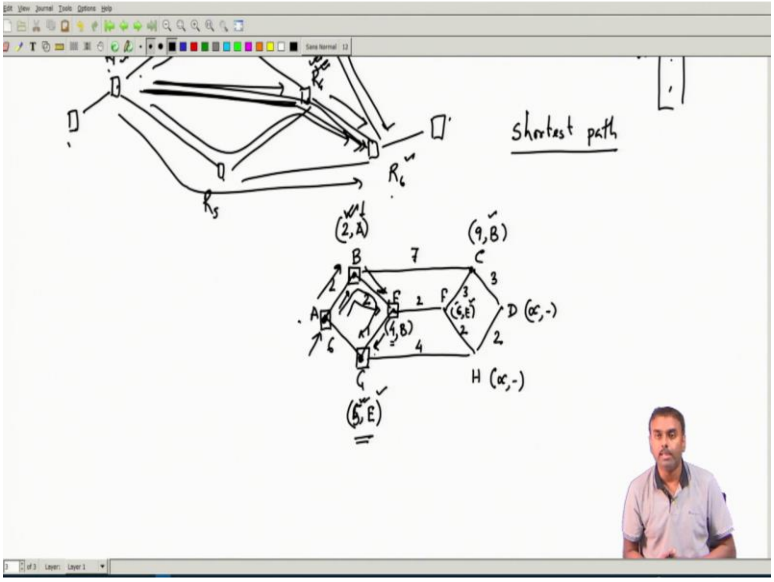
And then probably we will go to the original shortest path algorithm that has been implemented in internet protocol. So, we will also talk about those things if time permits link-state protocol for routing we will talk about then, and we will talk about this Bellman-Ford algorithm also. So, we will see how far we can go, but at least what we will try to do, we will try to see what is the algorithm instead of going into the because all shortest paths are shortest. So, it is an optimization problem.

So, we will have to define a proper optimization problem, how to define the shortest path, and then how I solve this optimization problem in a very big network, with a lot of nodes and a lot of edges, that is an optimization problem we will not go into the details of that optimization problem how that can be optimally solved what is the algorithm.

So, we will just give the algorithm that has been proposed without proving anything about why it means what kind of complexity it holds, and why it is optimal even though it is we will see that it is greedy locally greedy, but why it is optimal. So, all those things will not do, but we will this course is not for that, there should be a separate course in network optimization.

But in this course, we will just try to give the essence of what is the algorithm ok. So, let us try to see we will take an example and then try to demonstrate.

(Refer Slide Time: 21:10)



So, let us see a particular network; let us say this is the network. So, its nodes A, B, C, D, E, F, G, and H ok. So, whenever you are trying to do the shortest path of course, this link must give some weight value. It might be distance, it might be a delay, whatever the metric is according to which you are trying to calculate the shortest path, but that value should be given, then only your shortest path algorithm will come into the picture. So, this link weightage is very important.

So, let us take the link weightage to 7, 2, 6, 1, 2, 4, 3, 2, 3, 2 ok. Now, in this from node A, starting from node A we will try to calculate the shortest path to every other node, that is what Dijkstra came up with. So, we will try to see his algorithm ok we will just see the algorithm why it is optimal, what its complexity of it, and why it is efficient than others, we will not talk about that. We will just try to see how this algorithm functions actually, and that that part only will be covered.

So, what it does do from the starting node? So, it has steps, so we will be describing those steps. So, from the starting node A, it will try to see what other next neighbor he can connect to. So, as you can see A can connect to B and G, now you try to actually give the cost that it incurs to go from A if you take that path. So, suppose from A you go to B so the cost is 2. So, you put 2 over here, and from where it is coming, the immediate previous node.

So, from A it is coming to B, with cost 2, over here it is cost 6 it is coming from A you write that ok. Now, these are the only two you can connect to. So, now, you actually make them, these are temporary nodes that you have visited now. So, only B and G could visit, among this temporary node some of the nodes you will be making permanent. So, what is the technique? You try to see so basically what you will be doing, for every node initially this cost you will put as infinity ok.

And because you have not explored them you do not know. So, you put a dash that I do not know from which node it has to reach. So, every other node will have infinity and dash, infinity and dash, and so on. So, this will be there for every other node, only these two nodes are now populated. So, among these nodes now you try to find out what is having the lowest number; that means, that node can be reached from A in the shortest amount of cost. So, that node you make permanent.

So, every pass you actually make one node permanent ok, whichever has the shortest cost. So, now, I have A and B two permanent nodes, let us try to see from there which other node I can reach. So, from A I know already G I could reach. So, G will have the same value, but now from B, I can reach as you can see E and C. So now, I have to update their cost and corresponding this thing, update their cost and the corresponding previous node.

So, B I can reach from B, and what is the cost of E? 2 plus 2, so that is 4, and C I can reach oh sorry, the first number will be 4, and then the second number will be B. So, the cost is 4 and it will be B, and over here the cost will be 2 plus 7. So, that should be 9 and it is B ok. Now, what are the next nodes that you have explored, G E, and B? Let us try to see among them what is the lowest, that now I will make permanent. So, now, E is the lowest so I will make E permanent, done.

Now, I will try to see from E whom I can reach ok? So, E what are the new nodes I can reach? I can reach F and I can reach also G if an earlier visited node which means it has the most updated, but if I now get I will also check, if I can now go from here the cost is 4; from A if I start via this path the cost is 4 and from here if I go further and reach G, I will be adding 1. So, 4 will become 5 so I will update this and I will change it so that no longer I have the shortest path to G via means A, but it is actually via this E. So, therefore, the previous node is E.

Now, as you can see slowly, I am now constructing this path. So, as you can see earlier I had A to G which was the shortest path with 6 costs. Now, I have found another path as you can see trace it back to E, previous node E to B, and B to A because B's previous node is A. So, A, B, E, G, that is giving me cost 5. So, therefore, that is the 1, from E any other node I can reach? As you can see, I can reach over here from C also because means sorry, I can reach F.

So, let us update the cost of F, what is that 4 plus 2? That is 6 and it's coming from E ok; right now, from F and B ok nothing else we can reach. So, this is where I will end and I will try to see this cost is 5, this is 6, this is 9, these are infinity so therefore, now G will become permanent.

So, now, from G I will try to see whom I can reach. So, from G to H and this is how it will continue after all the paths means I reach all the nodes or all the nodes are permanent, we will see that the shortest paths are formed and I will be able to trace them back, what is the shortest path for every node.

Like over here G has become permanent. Now, you can see from A to G the shortest path is this one, A to E the shortest path is this one and so on. A to B shortest path is this one ok. This is how from A to every other node I will be now getting the shortest path. If you continue this algorithm till the end you will see that all the shortest paths will be found.

So, that is the Dijkstra algorithm, it is a very simple algorithm, but it works in a scalable fashion it converges always because after means every time every iteration you are making one node permanent. So, as many nodes are there, that many iterations you will be doing. So, it converges guaranteed convergence and also, it is very fast, we can prove the complexity of the algorithm, we are not doing it over here that is for another course, but it is a very low complexity algorithm.

So, this is how generally for every node if you have the picture of the graph, you will be able to calculate the shortest path. So, this is something written as an algorithm if you run, every node will be able to calculate its shortest path to every other node.

So, for that the entire graph is to be known. So, how you can know this graph is basically every node can broadcast what is his neighbor to other nodes by seeing all these broadcast messages and means if you aggregate. So, for every broadcast, he will get the broadcast of the neighbors also, and neighbors will get the broadcast of his neighbor. So, like this, if you aggregate all these things, you will be able to construct the whole graph.

Once you know the whole graph and associated links then you can actually compute or run this shortest path algorithm and you will be immediately able to think about what should be the next hop, for a particular link I am trying to reach. So, everybody knows from him to whichever destination he is targeting, what should be the shortest path, and which direction he has to go.

So, A knows if I have to go to G I have to forward to B, B also knows if I have to go to G I have to forward to E, E knows if I have to go to G I have to forward to this. So, they know this immediately and they will be able to implement that. So, this is one other protocol. Next time what we will be trying to do there is another protocol that will be simpler probably.

So, we will start talking about the Bellman-Ford algorithm which is another way of determining the shortest path. We will talk about that, we will talk about pros and cons, and then link-state protocol most probably will not have time for that, but that is another protocol that you should learn because that is the protocol that is being used in IP mostly. So, we will talk about that later ok.

Thank you.