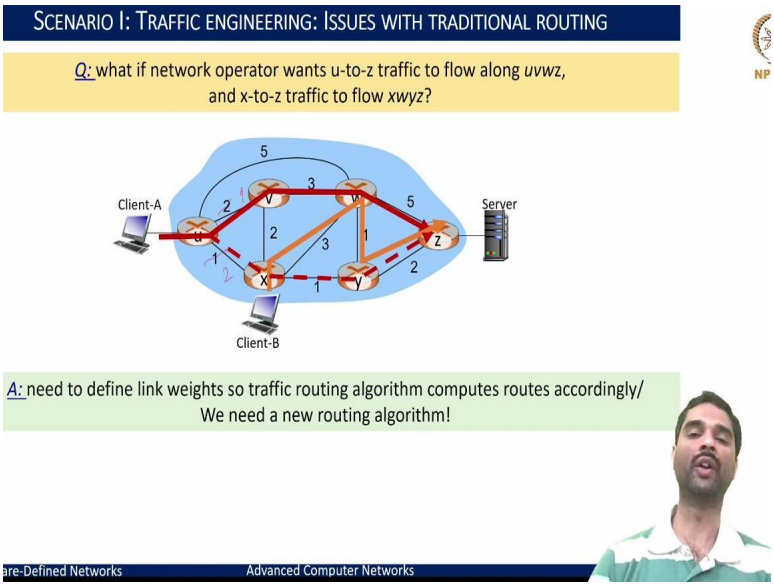
Advanced Computer Networks
Professor Doctor Sameer Kulkarni
Department of Computer Science Engineering
Indian Institute of Technology, Gandhinagar
Lecture 32
Software Defined Networking - Part 1

(Refer Slide Time: 0:17)

So, let us look at some of the fundamental challenges that we would encounter with traditional networking. This would help us to better understand and appreciate the role of SDN, and why we would need it.

(Refer Slide Time: 0:31)



Consider a topology as shown here with client-A, trying to interact with a server on the other end connected to the router z and the link weights trying to show potentially the latency of communication. Think of these weights as in inverse proportion: the lower the better. And if we had such a topology and let the routers be configured either with the link state or a distance vector routing algorithm and decide on what routing we would have when the client-A sends a packet to the server, we can notice if we solve it, and see that it would take the least cost path and the least cost path in this case happens to be u, x, y and z, the cost of four. And this seems a reasonable thing that we can predict in terms of the stable topology.

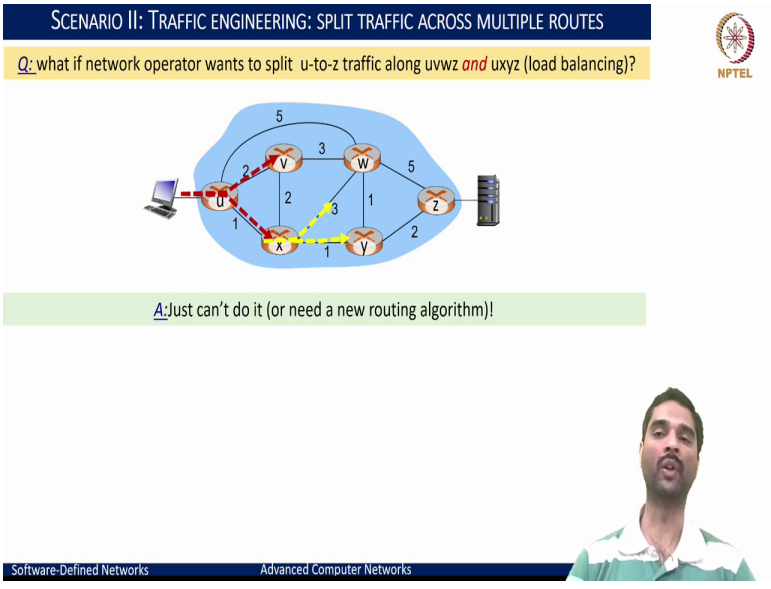
Now, suppose if we want to have a mechanism, or have a requirement, where instead of routing the flows from a client-A to the server, or the path of u, x, y z, we would want to tailor it and make sure that they go through the routers v and w on that to z. If this is the intent that a network operator has for this specific flow, now, the question is, is it possible to do this? And if it is possible, what is the thing that a network operator needs to do to ensure that this goal is achieved? And if we think about it now, we can see that if we are able to modify the weights of the links in such a way that the path u, v, w, z becomes the least cost weight, then the routing algorithm would automatically update itself and ensure that the path u, v, w, z will be taken into effect.

And we see this; what that means is now we would have to go on and change the link weights or in the minimum , let us say we swipe the link weights, then this would come into effect. But if we also have the route from Client-B going to the server, which also now needs to be tailored, then whatever the changes that would have made to come into effect for Client-A may not work directly for client-B, that means we would have to again re-compute and re-adjust the weights for all the links where the path from Client-B all the way to z that we need to make it operational. What this means is now just redefining the link weights as an exercise to get our intention of routing the traffic from one end to the other, through this particular set of routers may not work. And if we have to do this, it has to be done programmatically.

And that is where either we would need a new routing algorithm or a new mechanism to express our intent, so that the routing algorithm can take into account the aspects of our intent and then set up the routing record. So, this is a very fundamental challenge that we can see with the traditional routing.

And if we question ourselves, why is it, so we can see that because the routers take a decision on their own and do not have anything with respect to the network operators mode for a specific set of flows. There is no way that we can build this knowledge into the current routers, and that is the main concern here.

(Refer Slide Time: 4:29)



Next, let us look at another important aspect that we typically come across when we have multiple links is typically to split the traffic across multiple routes that we have. So, if we take again the same topology and we want to route all the way from the client connected to a router u or to the server connected to the router z, we can see how could we even split the traffic properly so that we are able to balance the load within the network. And this is a very typical characteristic that a network operator would be looking for to ensure the utilization of the network is at its prime, without over utilizing any of the links or under utilizing any of the network links.

And this would mean that we need a mechanism for the traffic that is flowing in at a particular router to be split across multiple paths in proper fashion. And this would now mean that you would want certain subset of the traffic that reaches u and destined towards z to go through a router v and a part of the traffic to go through the router x. And likewise, when the traffic has reached router x, you would want to split it and take certain portion of the traffic through the router w that is through the xw link and a certain portion of the traffic through the xy link to eventually make it to the destination z.

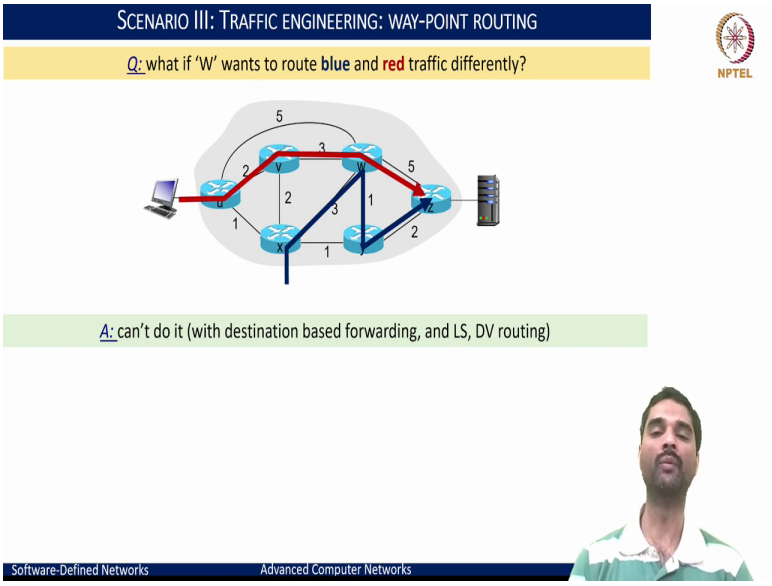
So, is this possible with the traditional routing that we have seen so far? And the answer is, it is just not possible. We would definitely need a lot more mechanisms or any other routing algorithm that would allow us to split the traffic in the way that we would want. And we would also want to have control over what should be the split of the traffic. Like if we say that the link weights are 3 to 1 that we would want to adjust the traffic in proportion to the link weights or an inverse proportion to the link weights or vice versa.

So, now if we ask ourselves again, like where is the problem and what are the limitations? why the traditional routings are not able to support this, the answer is quite simple. If we have a set of packets that are coming based on a certain destination IP, and our forwarding is happening with respect to the destination IP, then you are choosing one of the routes at each of the routers and set it up.

We can think of, we can have multiple rules for the same destination IP and set some proportions to say, I arbitrarily pick amongst the multiple rules and forward the traffic. But that could work but have unpredictable behavior in terms of which set of flows would end up taking what path.

And this is where we need to rethink how we could have addressed this specific problem of splitting the traffic across multiple routes.

(Refer Slide Time: 07:42)



Another key aspect that we see in the networks is what we call as a waypoint routing. And what this waypoint routing really means is to ensure that once the traffic reaches your campus or enterprise network, what you would want is to go through a certain set of middleboxes that we discussed earlier; like any of the traffic that is incoming, we would want to scan it ensure that it is safe to let the traffic in.

So, to do that, you may have to redirect the traffic towards a dedicated firewall that you would have set at the perimeter of the networks. And for any of the malicious or suspicious traffic that we may think we may also want to enable specific middlebox functionalities like deep packet inspection to ensure that we are really letting in just the safe traffic.

And we may have different policies for internal traffic like people within the domain, how they would access the resource, while people connecting to a VPN from outside the domain and ensure that we have the right safety requirements. And in such cases, you cannot deploy these middleboxes at every place, they would be at just one particular point in your network. So, any traffic that enters would have to go through these specific points to ensure that you are able to meet and treat these packets with the desired middlebox functionality.

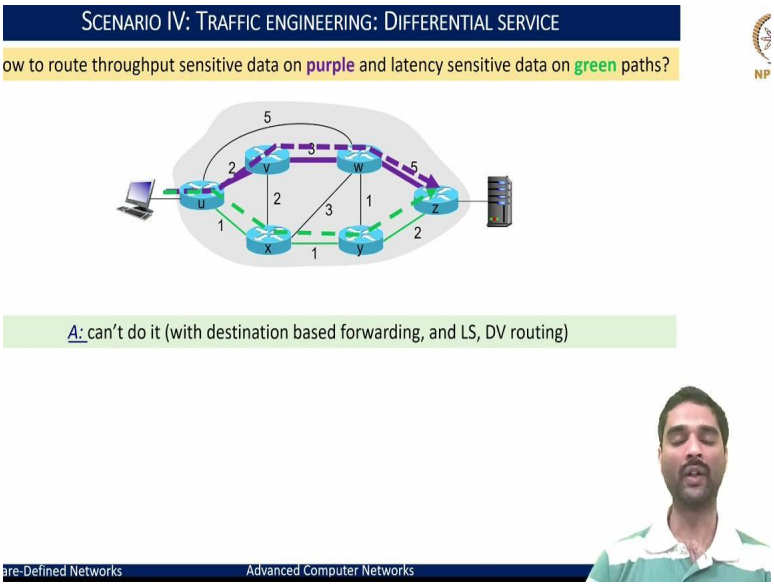
So, in this simple topology, if we consider that router w is attached with some specific purpose-built middleboxes, through which we may want to run our incoming traffic to go through, then we would require the specific aspect of how we would route any of the traffic that is incoming to go through this router w, regardless of the link weights here.

And what this also means is the traffic that is entering on u versus traffic that is entering on x would have to be routed differently. And if we say that the traffic on the right is the traffic external to the domain and traffic on the x is the traffic internal to the domain then we may want to have different functionalities achieved for these different middleboxes to operate on these flows. And in such a scenario now, if we again rethink of what is it that we have at hand to control and dictate this. Surprisingly, we have nothing that would ensure us to build this kind of framework.

And again, with destination-based forwarding, be it the link state or distance vector routings; we cannot govern or dictate that this waypoint routing be added. And this puts us in a lot of trouble in terms of how we want to build our infrastructure. And that was one that led many of the researchers to think of different means of how we could really ensure this waypoint routing. And now if we retrospect in this and see what is it that we would need to affect the waypoint routing, we can see that again, we cannot have just the destination-based forwarding.

But we would want the routing to consider the key policies as such in terms of if I say that blue traffic has a policy that it has to go through the router w and then to router y, eventually to router z, while for the red traffic, we have a policy, which says it has to go just through the router w and then eventually make its way to z then if we are able to tell this abstraction of providing these kinds of policies, and ensure that they can be met by the routing framework in terms of how the forwarding for a specific red packet and blue packets be dealt, then we would have achieved our purpose.

(Refer Slide Time: 11:48)



Let us look at one more fundamental aspect, which you have partly learned in the earlier series in terms of how to provide the diff service or differentiated services to different flows. And this is very common in the networks to say that we have several of the routes, but each of the routes may offer different bandwidth guarantees, and different latency guarantees and different loss aspects as well.

So, if we consider the simple topology that I have shown here, it says that the path that is shown on the purple is a very high bandwidth, considering the weights to be in proportion to the bandwidth that you would achieve. And the path that is shown on the green is a very low latency path and consider the weights now to be as latency, which would mean that the u, x, y, z is the least cost path in terms of the fastest path towards reaching the server, while the u, v, w, z is the optimal bandwidth path in terms of providing a better ability to provide better throughput for the flows to go through.

And as users, when we have different kinds of applications that we want to run, like if we have bulk file transfers to be done, we do not care as much about latency but we care about the throughput. So, even if it takes longer time for the files to go, if it is able to deliver a lot more traffic, that serves the purpose. Or if we think of traditional data centers, you will have a lot of the VMs that would be migrating the virtual files that would be backed up the database or that is backed up. All of these are throughput intensive. And in such case, what matters is the pipes with higher bandwidth.

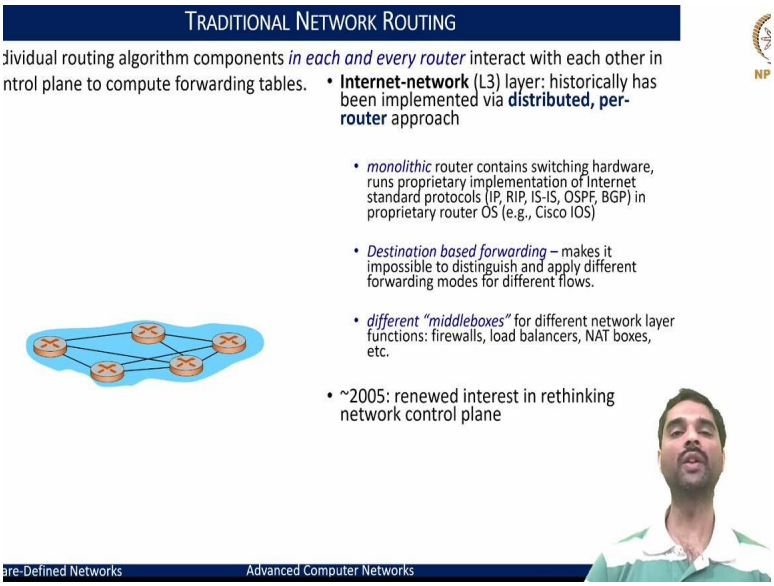
On the other hand, if we think of real-time streaming data, the VoIP calls that we have, even the stock exchange, like share market updates that you would want to do and receive, you would want to get the data as quickly as possible, the data or the content that you would want to transmit is much less, but you would want to ensure that the data is in real-time. And such kind of functionalities would require that we use paths that have low latency. And in this case, u, x, y, z as the green would fit the bill.

And if we consider these two provided by the same underlying network. Now for each of the applications, we would want to choose differently the paths based on the characteristics that it would demand. Now, the question is, is it possible to achieve this through the traditional routing? And the answer is plain, no, for the simple reason that we can at best build the destination-based forwarding for either of the routes, but we cannot guarantee which of the applications would end up taking which of the routes unless we augment and add additional mechanisms into each of the routers to say, like, when you look at the DiffServ field, you have to ensure that you are able to take a particular route for a particular path.

So, you have to have the backups of different paths and ensure that you are able to do that and not just limit your routing to be based on the destination IP, which most of the routers end up doing. So, until every router in the path is able to see that there is a means to look up not just at the destination, but also at the DiffServ field to ensure the DHCP code point and honor it, you would not be able to achieve it.

So, these found the fundamental challenges that we see when we look at the networks that are operated by different ISPs, the networks that we operate on campus or enterprise level. And we would need now the mechanisms or the right abstractions that we earlier thought about to provide us with facilitating these kinds of aspects. And that is where SDN really helps.

(Refer Slide Time: 16:12)



So, in traditional networking, what we see is each and every router learns on its own. So, we have lost control over what could be the eventual outcome of this learning and adapt it for our needs. And this distributed per-router approach is a concern. And moreover, these routers are in the sense monolithic, where they contain the switching hardware and proprietary implementations for these routing algorithms. And you may see that I cannot have in a given topology, one of them running OSPF versus the other running RIP. So, we would want all of these routers to be configured and managed in a way that they talk the same language. This is again another challenge when it comes to managing these specific devices.

Second, if we have these destination-based forwarding, it makes it impossible to distinguish and apply different forwarding modes for different flows. So, we need to overcome this model or philosophy of destination-based forwarding with something more richer abstraction that would allow us to build the right forwarding.

Third, we looked at the different middleboxes that would reinforce and affect the way we would want to build the forwarding plane. So, and also the functionalities like it is no more that the destination packet headers like when you have a NAT, it would completely alter the IP and the port that the packet is going towards. That means the destination IPs are affected. And even the load balancers would do the same, where they would select one of the back ends and plug a different destination IPs.

In such cases, we would want to have a generic approach to see how we would want to route rather than just relying on destination-based forwarding. And as we saw in the waypoint routing for these middleboxes to operate on the traffic, we would want to support these characteristics. And this is where the abstractions that we spoke about in the control plane were brought in, and this renewed interest led us to the redesigning of this network control plane.