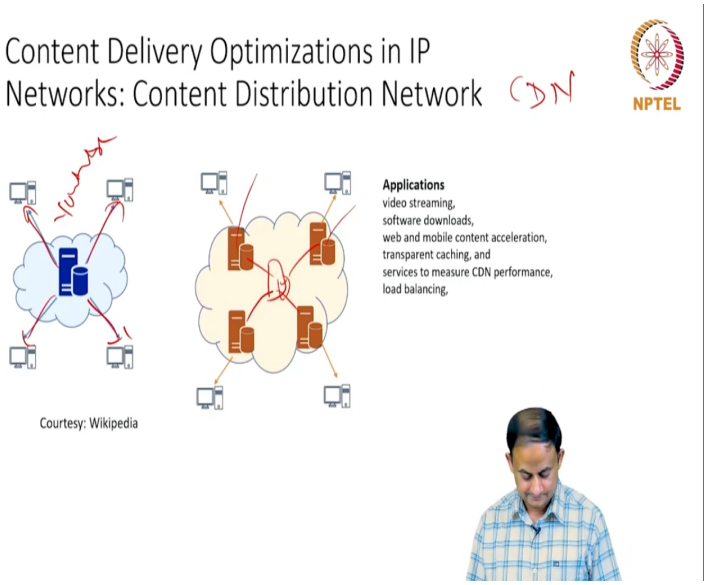
Advanced Computer Networks
Professor Doctor Neminath Hubballi
Department of Computer Science Engineering
Indian Institute of Technology, Indore
Lecture 62
Content Distribution in IP Networks - Part 2

(Refer Slide Time: 00:16)



And the third way of doing the optimization is something called as the Content Distribution Network, this in short form is also called as the CDN. CDNs are very popularly used in today's internet. So, let us try to understand what is CDN. So, let us say on the left-hand side what you have is one server.

So, let us call this Server maybe the YouTube server, and YouTube server has got bunch of videos. And what you see on the other side are the four computers which are downloading the videos from the server. So, this is how in point-to-point communication, the video content is actually distributed when there is only one server which is located at a particular location; maybe the YouTube server is located in the US.

But as they share distancewise, the number of the routers that are there in between. So, let us say the consumer is located in India, and the YouTube server itself is located in the US. So, distance wise the request has to go via many number of the intermediate routers, maybe let us say around 20 routers are there. So, the request also has to cross 20 routers, and then the response also has to

come back through those 20 routers, assuming the transmission is symmetric. Symmetric means the same route is taken for the request to go there and the response to come back. But this will also increase the latency of the transmission.

So, when the number of hops in the intermediate portion actually, increase then it takes a larger time because of the processing time at each hop. Because the router has to do the lookup operation and then do the transmission, and sometimes you sum up the portion of the net links that are congested.

So, you will have packet drops, many of the times, you need to retransmit the same content from the same source to this one. So, all these things come into the picture. So, in order to minimize the end-to-end delay in the transmission and also for the locality of the reasons what the YouTube server can do is to replicate the content, there is not necessarily one central server which is serving all the users which are across the globe; they can host a particular server in local region.

For example, there might be a server which is sitting in India which is actually serving Indian customers, or there may be a similar server sitting in some other country like Sri Lanka which is serving Sri Lankan customers. So, you might still have a central server and a central storage system, but that is actually giving content to all of these regional servers located in different geographies, and when the end users make the request, they are served by the local servers, not necessarily the central server.

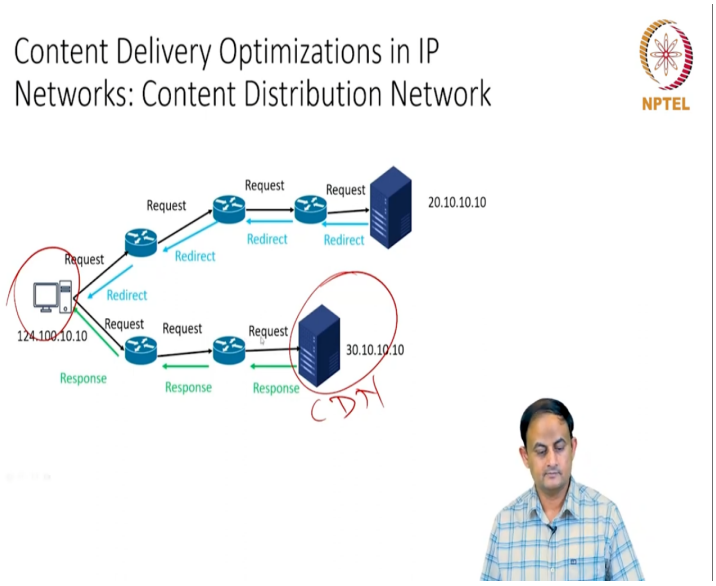
So, this is called the Content Delivery Network, and this actually helps to minimize the latency. The end-to-end is from the time the request originated from the end user to the time when the response actually arrives that is when the content actually arrives at the end user is actually reduced.

And in addition to that, it also brings a notion of fault tolerance. So, meaning if some of these other servers are down for some reason, the others can actually take the load. And in addition to that the content delivery networks also bring, because I have kept the server locally closer to the end users where the consumers are located, I can also bring localize the content.

So, for example, the Indian server, YouTube server can host content which is India-centric, whatever content which is developed by Indian content developers and likely to be consumed by Indian consumers or the users. So, that way, I can keep a different set of the content in this server and different kind content in the second server and so forth.

The main server, central server can still have the same content of all these replicated servers. But the local servers can still host the local content thereby you actually optimize. So, where exactly the consumer is, and what content is likely to be consumed by the consumers which are closer to that server, that content is actually being put, placed in that server, and then you serve it. So, that way, you actually minimize the number of times you go to the main server and access the content. So, this actually works well.

(Refer Slide Time: 05:36)



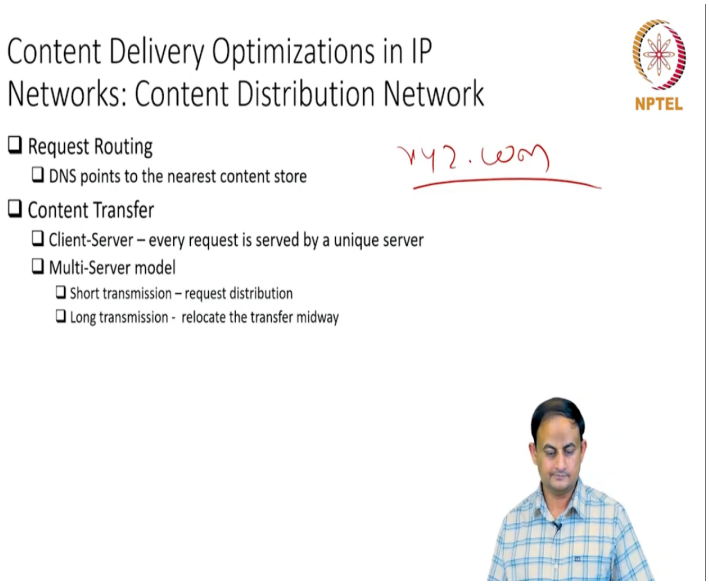
So, as an end user, when you make a request, you probably hit the main server, the main YouTube server think of this as a YouTube server. So, the first time you make a request, the request actually goes to the main YouTube server, and then the main YouTube server actually redirects you to the nearest point where the replica server, is actually located, and the user will establish a connection to that server, and then the content is actually transmitted back to the end user.

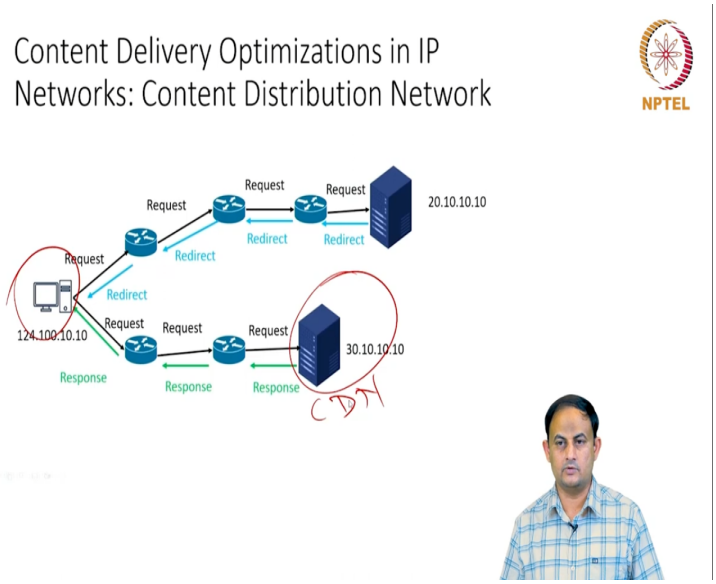
So, when you make the request, the main server can still take that request, and then okay, you are coming from so and so particular region, you are coming from India, there is a server located in India, why do not you access the content from the Indian server, actually, this is a CDN server located in India, and that actually gives you the content.

So, this is how actually CDN will work. So, this minimizes the number of the hops because the server is located in India, and the number of intermediate routers through which the request and response have to cross is minimized. So, because it is local only, so, instead of crossing the 20 hops, you might do it within 2 or 3 hops or, at the max maybe 4, 5 hops, you will be probably able to reach that server and then get the content.

So, that is how the CDN networks actually work. So, this is another way of minimizing the transmission. We are still operating using point-to-point communication. This guy is still establishing a TCP connection to this server, and then over that TCP connection, the content is actually being transmitted.

(Refer Slide Time: 07:36)



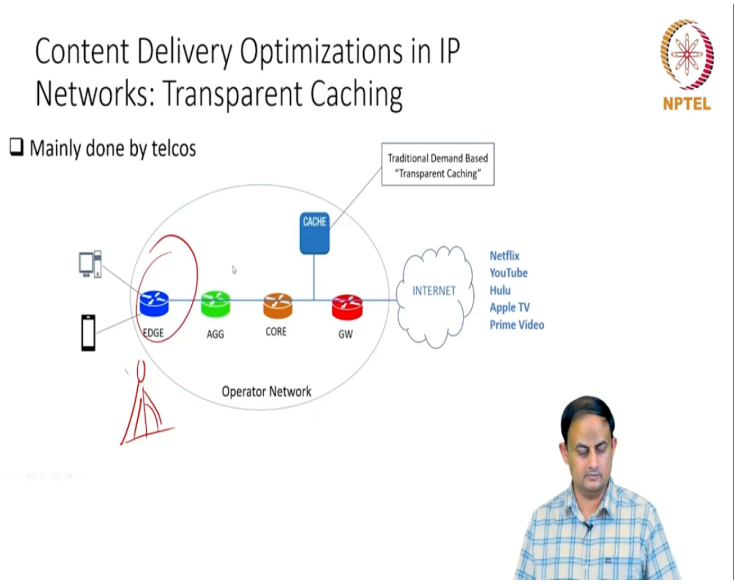


But all that it is doing is, it is actually minimizing the number of hops that get through which the content is actually flowing through. And another thing that you can do is you can tell instead of main server, in this case redirecting you to a server which is closest to the consumer, when the user makes the requests; let us say I open my web browser and then type xyz.com, there is a DNS query that gets generated to resolve the IP address corresponding to this domain.

The DNS server itself can notice the source IP address, from where the DNS request is coming from, and based on that, it can decide to give the IP address of that server which is physically closer to the end user. So, now the job of locating the server is outsourced, or finding a closed server is outsourced to the DNS server rather than to the main content provider server itself.

So, that is another way with which you can actually optimize the content distribution in the network. So, likewise, every request is served by probably a unique server. So, you have a pool of the servers, and then out of them, one of them which is physically closer to the end-user or which is actually lightly loaded at this point of time, will be chosen, and through that, you actually do the transmission to the end user.

(Refer Slide Time: 09:16)



And in addition to that, sometimes, particularly, the telecom companies also do an estimate of what is the content, which is the popular currently in the network, which is actually, let us say, there is a cricket match that is going on today or the match has ended and there is a particular portion of that event, game which is actually very exciting and somebody has hit six sixes in an over or somebody has taken a hat trick, three wickets in three balls and the clippings of those videos can become quite popular or some other time some event is happening. So, for example, something, a prime minister is visiting a particular city, and then video clippings of the prime minister's speech might go viral, and a lot of people might be interested in watching that. So, for the network service providers also, it becomes very problematic, they need to transmit the same content several times over their network. So, for example, if BSNL or Bharti Airtel is there. So, their networks or over their links and their routers, so, same content is actually being transmitted repeatedly.

So, what these people, particularly, if they are operating the mobile towers and the last mile connectivity, so, the end users might be connected to a mobile tower, so, somewhere closer to the end user, that is where the end users are located, if they get to learn that certain content is becoming popular, so, they might push the content closer to this base station, somewhere here in the network, by keeping some kind of storage in the vicinity of the base station.

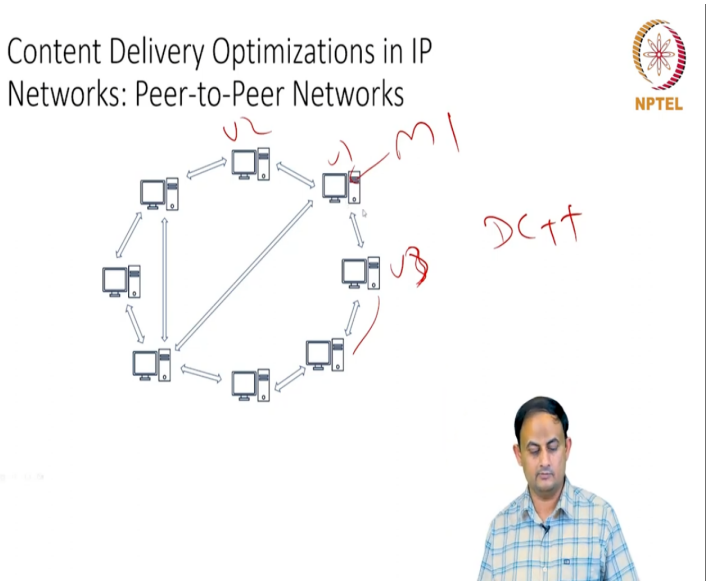
So, they can serve when the next user located or connected to the same base station, asks for the same content, same video clipping. So, the content is actually served from the repository, which

is actually located closer to that base station. So, thereby they also minimize the number of times the same content is being transmitted.

But now, this brings us to, now you need to understand what content is actually becoming popular and how do you estimate the popularity and stuff like that. So, assuming that is there. So, you can actually do only to some extent, not to the full. Every popular content, you cannot store it, but at least some of them you are able to cache and minimize the transmission.

Then it will reduce that much load on your links and the network. So, you can keep a transparent cache, the end user does not get to know, but the content is being served from the local repository, thereby you actually optimize the transmission. And the telecoms actually do this kind of distribution.

(Refer Slide Time: 12:27)



And the other mechanism that is popularly used in today's TCP-IP network is something called as peer-to-peer network. This you can think of as the overlay network. So, let us say I have a movie, or let us call that movie as M1, and some user which is part of this network, the first time he downloads that movie and if let us say the other users. So, this is the user 1, user 2, and user 3; when user 3 and user 2 ask for the same movie, instead of going to the server and then fetching the content, what this overlay network does is it learns that the copy of the same movie M1 exists in with the computer of the user 1 and he is closer to me, why do not I download from

the user PC of that user and when user U2 and U3 get a copy and they can in-turn distribute that copy to the other users.

So, some users are randomly connected to other users depending upon how the physical connectivity and whether those users are active or not active at this point of time, there is a structure, there is a connectivity that exists among the users of this network, and they actually exchange, they pass on that file which movie from one user to another user.

There are a bunch of such examples of peer-to-peer networks. So, for example, DC++ is one popular application. There is something called e-donkey, and the BitTorrent are all examples of such peer-to-peer networks. So, basically, from one user computer to another user computer, they are actually transmitting the content.

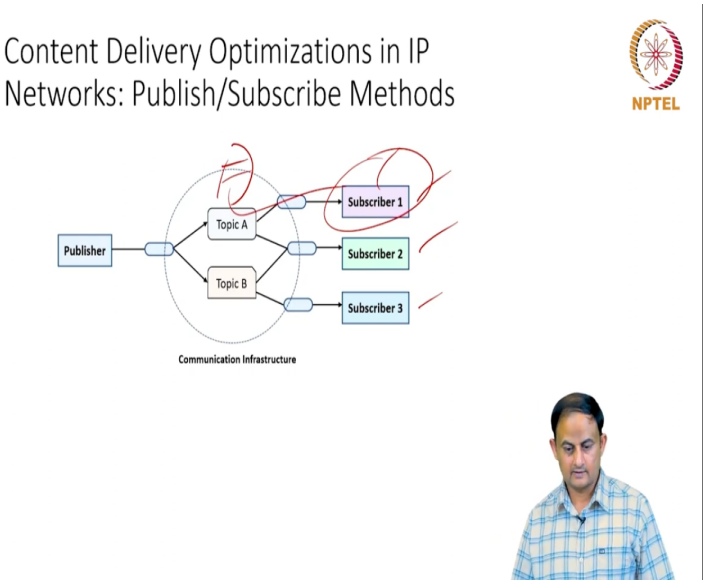
So, even if one or two users are not active this time or they leave the network, this overly structure or peer-to-peer network knows how to regroup and then maintain the connectivity among the people who are left over in this network and then do the distribution among themselves.

So, this also reduces the burden on the main server. So, if the movie M1 is kept somewhere in a central server like the example that we took, maybe the YouTube server, then that server within, even if you use the CDN servers as well, replicas as well, so, only a handful number of the servers are actually responsible for doing the transmission.

But unlike that, what is happening here is everyone is acting as a server, and everyone is also acting as a consumer. So, if I want some content, I can ask my neighbor; if they do not have it, they can also point out to their neighbors, and then I do not have it; now, this guy has, why do not you download from this guy?

So, they can coordinate among themselves and then bring the content to the required place, or the required user can access it from one of the nearest ones, whoever is active at this point of time. So, this is also another way of bringing the content closer to the user without actually going to the main server.

(Refer Slide Time: 15:42)



There is something called as a Public Subscribe Model for the distribution. The way this works is let us say, there are n number of users. So, here I am calling it a subscriber 1, 2 and 3. Here, there are three subscribers. And some users are interested in some particular type of the content, maybe the let us say, the subscriber number 1 is interested in the news related to the research activities happening currently in so and so places, the subscriber number 2 is interested in all the content that is related to the sports, maybe the cricket, tennis or other sports related stuff and the subscriber number 3 is interested something else.

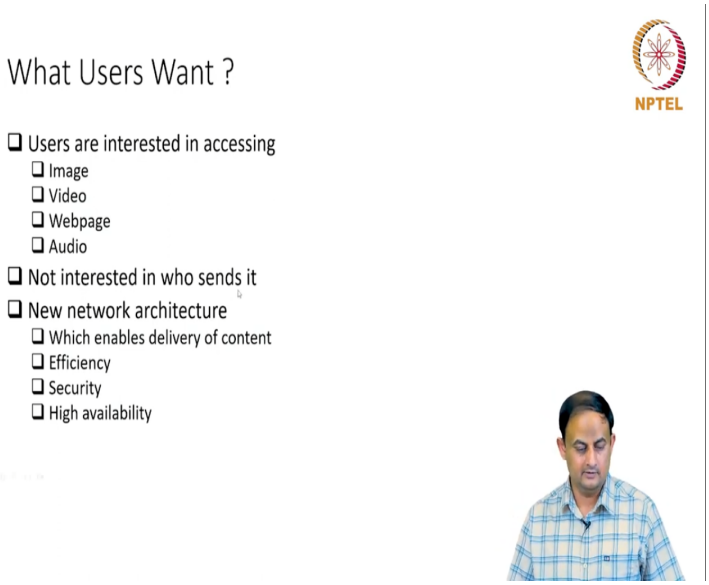
All the individual subscribers, are interested in a subset of the total content being generated. When the content gets generated, there is a publisher, and he tags the content with one of the topic names. So, maybe the topic name is sports, the topic name is scientific discoveries, topic name is something else, maybe tennis or something like this, and every content is tagged with that particular topic, and the subscribers actually show interest in a particular type of content and because of networking issue, there is intelligence built into the network.

There is a server; there is a registration mechanism and all that. And the network job is to efficiently distribute the content or deliver that particular content. If subscriber 1 asks for the sports-related content, the network brings that sport-related content and delivers it to subscriber number 1.

So, whoever is showing interest in whatever content type, that content type or the data is actually getting delivered to the subscriber. So, the first models are also available in today's TCP/IP network. But again, using the same point-to-point communication mechanism, we are bringing a retrofitting, a mechanism, or a delivery mechanism on top of this TCP/IP communication model.

So, for example, if I put one kind of server here which is actually keeping the content with the tag names when subscriber 1 asks for I am interested in receiving this, whenever subscriber number 1 comes up when he joins the network, it figures out, okay, he subscribe to so, and so type the content I have a new content of that particular tag name, let me do the transmission to this particular user. So, again, you still establish the same TCP connection, you use the same IP addresses, and end points, and then you do the transmission. But all that is happening is you are not transmitting blindly the same content to everyone else, but only to a limited number of people.

(Refer Slide Time: 18:43)



So, that is how content is actually getting transmitted. But what exactly the users are looking for when I am doing and I am saying that you are actually bringing content closer to the end user. You are actually putting some alternative servers, the replicas, I am caching the content to what all of these are actually doing. Some of them are transparent to the end users.

So, what the user is interested in is he wants a particular image to be downloaded, whether that image is coming to his computer or not, whether he is able to see that image or not. He is interested in a particular video clip whether that video clip is coming to his computer or not. He is interested in reading a news page or the page of news, page of a particular web server whether that is actually coming to his computer or not, whether he is able to read that or access that web page or not. He is interested in some audio communication, whether that audio is reaching to him or not. So, user really does not care where the server is and who is actually serving that content. I am not at all when I am reading or watching a video clip of a particular sports event, some 2 minutes of video content, I am not at all interested in whether a guy who is sitting next to me on the left-hand side gives me that video or the guy who is actually sitting right to me and he is giving that content, or it is coming from a thousand miles away, somebody or someone who is a thousand miles away, all that I am interested is in getting the content, what I am interested to watch at this point of time.

So, using the point to point communication, as we said if the guy who is actually having the content in which I am interested is a thousand miles away and if my neighbors are also interested in the same content, because of the point-to-point communication, this actually becoming inefficient transmission. So, can we bring some kind of efficiency to the transmission?

Although we did discuss some of the mechanisms that are retrofitted into the TCP/IP model of the current connectivity they are having certain issues. So, the security is an issue, the availability of the content, for example, if the server which is a single server that is sitting at one point of one place in the network, if that server is down for some reason, one or the other reasons, all the users who are dependent, who are interested in the same content will not be able to access it.

There is a single point of failure. Of course, you can bring fault tolerance by replicating the content, keeping multiple servers, and all that. But can we do that on a larger scale? Can I do the replication? So, although the content delivery network, and the caching that we talked about, all of them are essentially doing the replication.

But can we take this replication to another level? Can we bring the efficiency of the distribution to a larger extent? The CDN, you do the replication to a limited extent, you are bringing the efficiency.

But now we want to take it to the next level. So, another question that we are interested in when I am downloading a movie when I am accessing certain content, the notion of security should be inherent to the content itself, not necessarily to currently what has happened, TCP or TLS connectivity is actually giving you a kind of the tunnel that is establishable to two endpoints.

So, whether I connect to the main server or to the replica server, I still establish a TCP connection, a tunnel is established over which the content is actually transmitted. So, can I make this work the other way? I do not want to establish this connection to the server, but what I want is when certain content is actually getting transmitted over the network, the security is assigned to that content itself rather than talking in terms of the end-to-end tunnel.

And in the process, if you are able to replicate the content to a larger extent, you also bring the notion of high availability. So, somebody shutting down will not likely take down the content available in the network, at least in the immediate vicinity of the time. So, that is what we are interested in.

(Refer Slide Time: 23:42)

So, in effect, to summarize, the content delivery in today's network, there are many, many techniques, particularly, keeping the notion of the point-to-point connectivity of the TCP/IP model; we have built several ad hoc mechanisms to do the distribution, bringing some notion of the efficiency. No doubt, they have brought the efficiency, but we want to take it to the next level.

And when I talked about this replication be it the proxy server, be it the caching server, be it the CDN servers, all of them are doing the replication. But there is no unique standard that actually integrates all of them. So, not necessarily, all of them, in the sense, has one CDN network.

So, YouTube has its own CDN infrastructure, and some other OTT platforms might have their own CDN infrastructure, and they do not know how to talk to each other. So, only the servers which are maintained by the service provider of one company would be able to bring the optimization to a limited extent for their own content and for their own customers, not across all the customers, not across for all the content.

And how about bringing this optimization and extending it to all the customers, irrespective of what they asked for, from where, and what kind of content they asked for? That is the question we want to ask. And this operation, many of them, as I said, are controlled by their own service providers; they are not able to talk to each other.

And even if you maintain the local replica of the content, whether it is caching or the CDN, keep the content in the CDN servers, the data consistency issues might pop up. So, if your main server content is updated, the replica, we need to pass that content from the main server and keep it updated.

And now, the question is how quickly can I do that or how many times, how quickly I need to go and pass that content, and how do I actually know that something new has been updated on the main server, which is of interest to me. So, there are mechanisms, but we want to take it to the next level.

And as I said, the security itself should be given to the content itself, not to the, I want to establish a tunnel between point x to y and over which the content is actually getting transmitted. So, these are some of the limitations and also serve as the motivation for having the Information Centric Networking.

So, with that background in mind, let us go and understand how exactly these things are addressed in the Information-Centric Networking. So, that is the content or discussion agenda for the next lecture. I will stop it at this point of time. We will come back and see that in the next lecture.

Thank you.