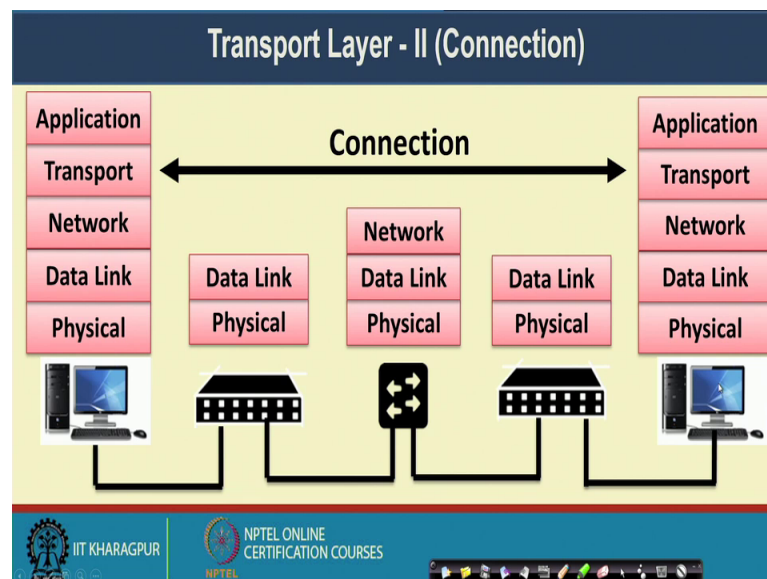


Computer Networks and Internet Protocol
Prof. Sandip Chakraborty
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 12
Transport Layer – II (Connection)

Welcome back to the course on Computer Networks and Internet Protocol. So, we are looking into the different functionalities of the Transport Layer of the protocol stack and in the last class we have looked into that what different services the transport layer can provide on top of your unreliable datagram delivery that is supported by the network layer. And what we have seen that the packet delivery the end to end packet delivery at the network layer is unreliable. And the transport layer provide certain end to end services on top of that. So, from today onwards we look into the details of all those services which are being provided by the transport layer.

(Refer Slide Time: 01:00)



So, the first service that we are going to talk about is about the connection establishment. So, as we are looking or discussing in the last class, that the 2 end of the devices which has the entire 5 layers of the protocol stack. So, that 2 end need to first setup a logical connection between themselves. And this logical connection is something like that one person is saying about hello and another person is replying back with another hello

message. And they are they establish a logical link among themselves and they both of them become sure that they want to share the further information among themselves.

So, this connection establishment is to see that whether the other end of the communication is alive or not whether that is ready to receive the message or not. And if it is ready to receive the message if it acknowledges then we can safely start sending the data. So, in case of your voice network like the telephone network you can just do it by saying hello. Because you know that it is a circuit switching network and whenever you are saying hello the packet will always or your message will always reach at the other end the reliability is not an issue here.

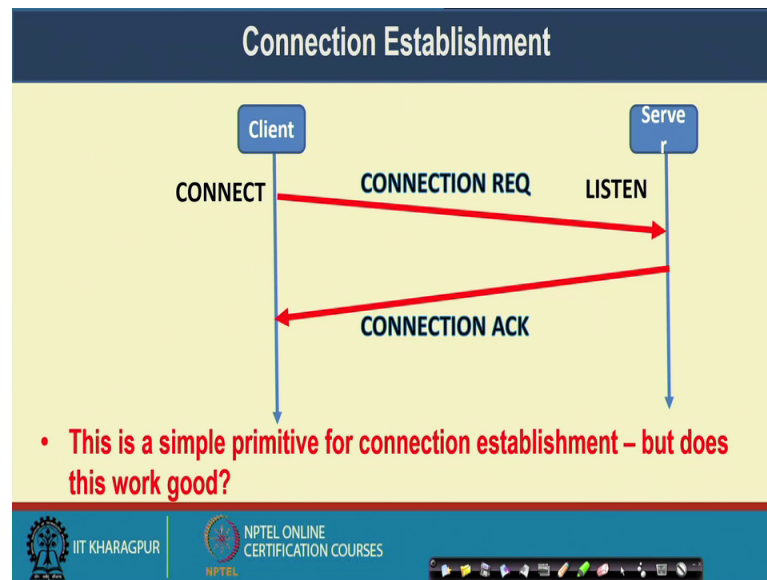
But in case of a data packet switching network, this reliability is an issue because this entire packet switching network is working on the basis of a store-and-forward principle where as I was mentioning the last class that every intermediate device has a certain fixed amount of buffer and whenever you are putting certain packets into that or certain data into that and if your network load is too high, it may happen that the buffer becomes full and packet starts getting dropped from that buffer.

If it happens then it becomes difficult for you to understand or to ensure that whenever you are saying hello, whether that message is correctly being received by the other end or the second scenario can be like the other end is not ready to receive your message and that is why it is not echoing back the hello message or not acknowledging your hello message.

So, that is why ensuring this logical connection at a packet switching network, for data delivery, is a little bit non-trivial compared to what is being used in case of your traditional circuit switching network or in the telephone network. So, we will look into the different aspects of this connection establishment, in the context of transport layer of the TCP/IP protocol stack, that how you can ensure that whatever hello message you are transferring to the other end the other end is correctly receiving that hello message and correctly been able to decode that hello message. And it is able to send you back with the required reply.

So, let us look into the connection establishment in details. So, the connection is just like a logical pipe that ensures that both the ends are now ready to send or receive further messages or further data.

(Refer Slide Time: 04:20)

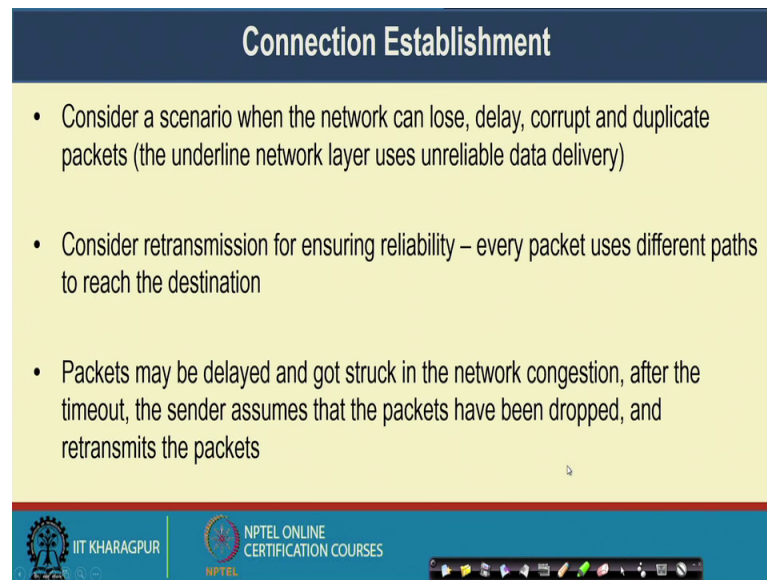


So let us see a very naïve protocol of the connection establishment. So, we have in a we are in a client server model. So, in this client server model or the client is trying to make a connection with the server.

So, we say that the server is in a listen state, the server is listening for the incoming connection. So, the client sends a connection request message. So, once the client send the connection request message the sever is in the listen state. So, the server can listen that connection request message and it replies back with the connection acknowledgement message. So, this 2 way hand shaking is likely to work for a normal connection establishment purpose, but our life is not very simple in case of a packet switching network.

So, the question is that this simple primitive where the client sends a connection request message and the server responses back with the connection acknowledgement message. Just like the hello protocol that we use in case of our telephone network, whether that will work in case of packet switching network or data network or not. So, our target is to look here, that this simple primitive for connection establishment whether this will work good for a packet switching network or not.

(Refer Slide Time: 05:49)



Connection Establishment

- Consider a scenario when the network can lose, delay, corrupt and duplicate packets (the underline network layer uses unreliable data delivery)
- Consider retransmission for ensuring reliability – every packet uses different paths to reach the destination
- Packets may be delayed and got stuck in the network congestion, after the timeout, the sender assumes that the packets have been dropped, and retransmits the packets

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

Now, the problem in the packet switching network is that, the network can lose the packet there can be packet lose from the network there can be arbitrarily delay in delivering the packet. There can be delay in delivery the packet because it may happen that the intermediate router switch are there that intermediate routers their buffer is almost full and it is receiving packets from multiple other links and it need to transfer the packet one after another.

So, just like a scenario in a road congestion. So, whenever a road become congested then the speed of the cars becomes very slow. And all the cars are going to enter to a common road from multiple others road and in the road junction because it has a finite capacity, that becomes the bottleneck and the congestion becomes that there because of switch the speed of individual cars become very slow.

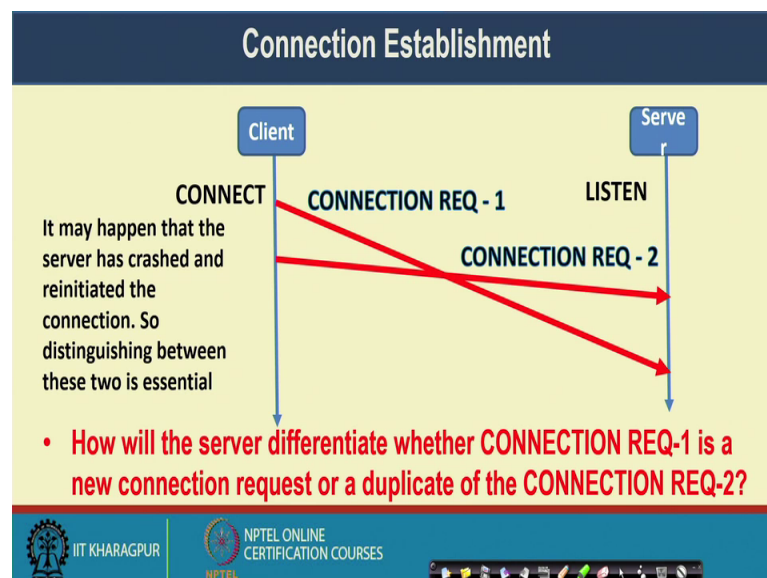
The same thing can happen in a computer network because a router is receiving packets from multiple other neighboring routers and when it happens it may in pay result in a congestion in the network, because of which the rate of packet becomes very slow. And that is why there can be this kind of arbitrarily delay in the network. The packet can get corrupted as well and there is a possibility of duplicate packet delivery. Because the transport layer also ensures reliability and the way of ensuring reliability in the transport layer is just like to monitor whether a packet is being received by the other end or not. If the packet is being received then I am happy if the packet is not being received if I am

able to find out that the packet is not being received, then what I will do that I will return with the packet after a time out.

Now, it may happen in the network that well the earlier packet that I have transferred, that package got stuck somewhere in some intermediate queue in the network because of the congestion or this kind of network effect. And I am keep on waiting for the acknowledgement and I do not get the acknowledgement within that timeout duration. So, I think that well the packet is probably got lost and then I retransmit the packet again, but whenever I am retransmitting the packet again note that the earlier packet was actually not lost rather the earlier packet was just waiting in a queue to get it deliver.

So, because of this reason it may happen that well the other end the receiver may receive multiple packets of the multiple or better to say multiple copies of the same packet which we call as a duplicate packets. So, it may happen that there is this kind of duplicate data delivery in their network because of this retransmission to ensure reliability. Now as I have mentioned that because the packet may get delayed done got stuck in the network due to congestion, the sender assumes that the packet has been lost it retransmit the packet and that way the receiver can get the duplicate packets.

(Refer Slide Time: 08:52)

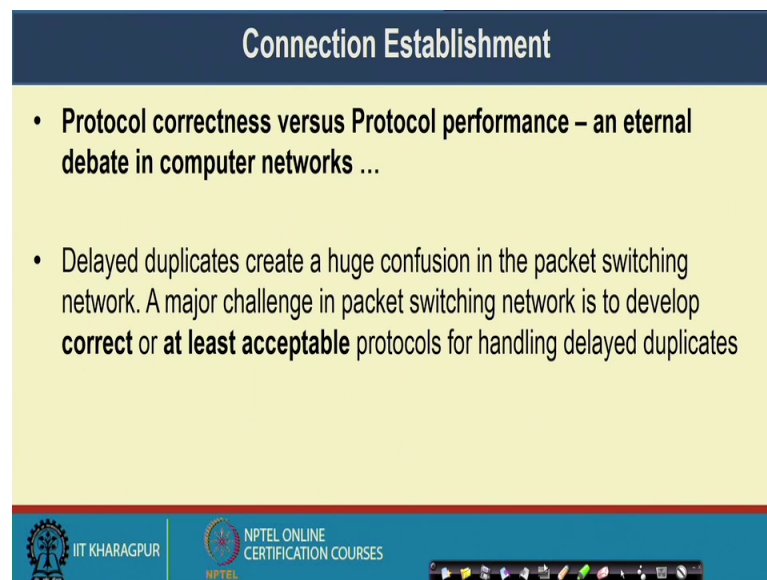


Now, when it happens, you can think of scenario like this. That well now the server has received 2 copies of the connection request. So, it has received one connection request here. But remember that this particular sequence number is not there, in the original

packet this is just to give you an indication that well there are 2 different connection request packets. So, the server has received one connection request packet and then it has received another connection request packet. It may happen that this particular connection request packet got delayed and it was transferred by the intermediate router after sometime. Because of that delay it has received late compared to this fast connection request packet.

Now, the problem for the server is to find out that whether this connection request one that it has received, whether that is a new connection request or it is a duplicate of the connection request to that that it has already received. Now the interesting fact here is that, it may happen that the server has crashed and reinitiated the connection so, distinguishing between this 2 becomes very difficult that, whether it is just like new packet new connection request message that is being received or it has happened that well either the server or say for this example the client has crashed after sending this first connection request packet and then the client is trying to establish another connection request.

(Refer Slide Time: 10:31)



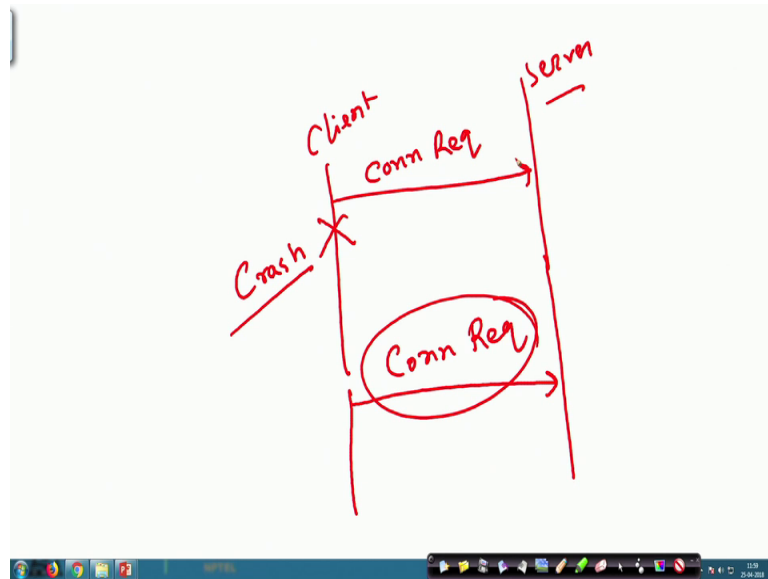
Connection Establishment

- **Protocol correctness versus Protocol performance – an eternal debate in computer networks ...**
- Delayed duplicates create a huge confusion in the packet switching network. A major challenge in packet switching network is to develop **correct or at least acceptable** protocols for handling delayed duplicates

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, even if you forget this particular scenario it may happen that it may happen that well.

(Refer Slide Time: 10:40)



So, it may happen that say here is your client and here is your server. So, the client has sent one connection request message after the client has send that connection request message see at this point the client has crashed. So, there is a crash here. So, the client got crashed and after some time, the client again reinitiates and it is sends another connection request message to the server.

Now, when the client sends the second connection request message to the server, it becomes difficult for the server to find out whether this connection request it is a new connection request or it is a duplicate of this connection request. Because remember that the server does not know whether the client has been crashed or not that information has not reach to the server. So, because of all this reason, the entire principle of connection establishment in a packet switching network is very difficult, because you need to differentiate between the original request and it is delayed duplicates and the challenge comes that how will you differentiate between the original request and the corresponding delayed duplicate.

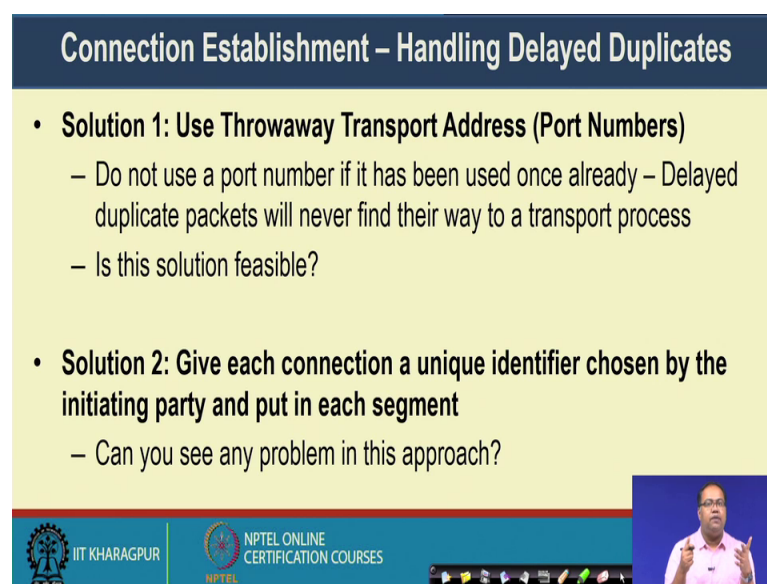
So, in the context of correction establishment, we always has this kind of debate. That whether we will go for the protocol correctness or we want to design a protocol which will perform good. Because if you want for the correctness what you have to ensure that you need to add multiple other modules to differentiate between a new connection from a delayed duplicates.

So, the question comes that whenever you will execute those modules for finding out whether that is a old connection of or a delayed duplicate message or a new connection request, this entire protocol things become complicated and it reduces the overall performance because this works like a over head for the data delivery.

You are not actually doing the data delivery that rather you are spending a considerable amount of time just for establishing the connection. So, that is why you have this kind of debate on whether we want a correct protocol or whether we still can were go it to a compromised, little compromised protocol which is not totally correct it can fail under certain scenario, but still it will give good performance. So, this delayed duplicate they create a huge confusion in the packet switching network.

So, a major challenge in a packet switching network is develop a protocol which will be able to handle the delayed duplicate. So, it is just like that sometime, we design a protocol which will completely be able to handle the delayed duplicates. So, you will give the preference over correctness or some time we give preference over performance. And whenever we give preference over performance still we need to find out a protocol, which will have at least acceptable level of conformation in handling the delayed duplicates in the network.

(Refer Slide Time: 13:47)



Connection Establishment – Handling Delayed Duplicates

- **Solution 1: Use Throwaway Transport Address (Port Numbers)**
 - Do not use a port number if it has been used once already – Delayed duplicate packets will never find their way to a transport process
 - Is this solution feasible?
- **Solution 2: Give each connection a unique identifier chosen by the initiating party and put in each segment**
 - Can you see any problem in this approach?

The slide footer includes the IIT Kharagpur logo, the text 'NPTEL ONLINE CERTIFICATION COURSES', and a small video inset showing a man in a pink shirt speaking.

So, let us see what are the different possible solution that can we that can have in this context. So, first of all you can use throwaway transport address or the port numbers so,

we have discussed this earlier that this port number it is a mapping between your transport layer and the corresponding application. So, it may happen that multiple applications in your machine are trying to use the TCP protocol to transfer the data. So, it is just like that, you have this application 1 and application 2 which are running on a machine and both of them are transferring data. Now whenever your network protocol stacks this is a transport layer of protocol stack, whenever it receives some data from a remote host it need to find out whether that particular data is for application 1 or application 2.

So, during that time we use the concept of port number, to differentiate between application 1 and application 2. So, this port number application once runs in one port say it is running in 8080 port application 2 runs in a different port say it is running in 2345 port. By looking into the port number in the transport layer header we will be able to differentiate between application 1 and application 2. Now all though will be able to differentiate between the application, but the question comes that can we utilize this port number to differentiate between the normal packet and the delayed duplicate. Now if we design a protocol where if a machine get crashed, it will use different port number for initiating a new connection. If that is the case, then probably we will be able to solve this problem.

So, it is just like that that our solution says that do not use a port number, if it has been used once already. So, if you have already used the port so, the delayed duplicate packets it will never find their way to a transport process. So, it is just like that say this application 1 say application 1, I am writing it an A 1. It was initiated a connection establishment message say port through port 8080 and after that this particular process get characterized. Now if you are running the application again then run it in a different port say 8082.

If it is the case and if you are sending another connection establishment message here, then this earlier connection establishment message that you have send through port 8080 whenever you will receive a reply of that say a reply of this connection establishment message that will also come in port 8080 and the transport layer will not be able to deliver that and it will correctly discard that particular reply message. And if a reply comes in port 8082 the reply comes in port 8082 then the transport layer will be able to deliver it to the application A 1.

So, this is a possible solution, but the problem comes that this solution is not feasible. Because we have a finite number of this kind of transport addresses of port number because we have this finite number of ports. So, you cannot throughout a port number once it is being used. So, in that case theoretically will be requiring infinite number of port addresses which is not feasible for the practical implementation point of you, and whenever also your utilizing multiple application. So, there are multiple applications which are kind to send data over the network.

So, the second solution can be like that give each connection unique identifier, which is chosen by the initiating party and put that unique identifier in each approach. Now this approach looks good, but the problem with this approach is that every time you need to design a unique identifier and you need to ensure that identifies is unique globally. So, ensuring that identifier is unique globally.

Again the problem is that what would be your algorithm to generate that identifier and even if you design an algorithm to generate a unique identifier, which will be able to sustain even after a system is getting crashed. You have to; obviously, use certain kind of hardware trigger here because you want to initiate that even after the system get crashed and recover from that crash, it will not use the old identifier that is being utilized once. So, that is why this particular algorithm also has a amount of overhead associativity.

(Refer Slide Time: 18:33)

Connection Establishment – Handling Delayed Duplicates

- **Solution 3:** Devise a mechanism to kill off aged packets that are still hobbling about (Restrict the packet lifetime)
 - Makes it possible to design a feasible solution

The slide features a background illustration of a network with blue nodes and green connecting lines. At the bottom, there is a blue banner with the IIT KHARAGPUR logo on the left and the NPTEL ONLINE CERTIFICATION COURSES logo in the center. On the right side of the banner, there is a small video inset showing a man in a pink shirt speaking.

So, the third possible solution that, we can utilize is to design a mechanism to kill off the aged packets or the old packets the networks. So, that is just like the restricting the packet life time. So, if you look in to the problem that we are facing it is because of the delayed duplicates. So the duplicate packet which have been transmitted earlier, but that got stuck somewhere in the network now those packets have been being transferred to the other end. So, whenever those have been transfer to the other end then the other end is in a confusion whether that delayed duplicate is just because the system has got crashed and now recover what I am sent a new packet new connection request packet or it is just delayed duplicate of the old connection request packet through which the connection as already been established.

So, if because all this problems our life, becomes complicated because of this delayed duplicate. If we can eliminate the possibility of delayed duplicate from the network, then this entire solution become simple. Now the question comes that how will be able to eliminate the delayed duplicate from the network.

And the solution is that if you associate with a packet life time with every individual packet that you are sending in the network, then you can say or you can design the protocol that well, once you are sending a new connection request message, you will make sure that the old connection request message it has already died of or it has already been taken out of the network, because its lifetime has been expired.

So, this particular solution 3, it makes it possible to design a feasible solution.

(Refer Slide Time: 20:19)

Connection Establishment – Handling Delayed Duplicates

- Three ways to restrict packet lifetime
 - **Restricted Network Design** – Prevents packets from looping (bound the maximum delay including congestion)
 - **Putting a hop count in each packet** – initialize to a maximum value and decrement each time the packet traverses a single hop (most feasible implementation)
 - **Timestamping each packet** – define the lifetime of a packet in the network, need time synchronization across each router.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

Now let us see that how you can designed it is solution. So, the first requirement is that you need to restrict the packet lifetime you need to design a way to restrict the packet lifetime. So, there are 3 different ways to restrict the packet life time. The first one is that you make a restricted network design; that means you prevent the packets from looping. You can have a maximum delay bound which also include the congestion delay on every individual packet.

And if a packet expires that particular time from it is originating time, then that packet is automatically drop from the network. The second is start second solution is that you put a hop count information in each packet. So, the idea is that whenever you are sending a packet in the transport layer in that packet you put a maximum hop count value say the maximum hop count value is 10.

Now, whenever a packet is being covers over the network then every individual hop just reduces that hop count. So, whenever it goes to the first a hop router it reduces it from 10 to 9. Whenever it goes to the second top router the second top router reduces it from 9 to 8 and that way it goes on. And whenever that hop count becomes 0 it will simply drop that packet. So, this is a very feasible solution which is in that used in today's network, to ensure that a packet is not hopping in the network for infinite duration.

The third possible solution is you put timestamp it each packet and that particular timestamp will define the lifetime of a packet. But this particular solution is not very

feasible or not very practical from a network perspective because in that case you require proper time synchronization among individual devices in the network, which is very difficult to achieve in a real scenario. Because whenever you have 2 different systems there will be a certain clock drift between these 2 systems.

So, ensuring this lifetime based on the timestamping of each packet where you will be requiring strict synchronization across different devices, ensuring that is a little bit different. So, normally go to the second solution that we put a hop count information at every individual packet and whenever the packet is being delivered by the network layer to the routing algorithm, at every individual router or at every individual hop it decrements that hop count value. And whenever it reaches to certain maximum hop when the hop count value becomes 0, during that time that the router if it receives a packet or receives a data packet with hop count value 0, it simply drops the packet.

(Refer Slide Time: 23:04)

Connection Establishment – Handling Delayed Duplicates

- **Design Challenge: We need to guarantee not only that a packet is dead, but also that all acknowledgements to it are also dead**

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

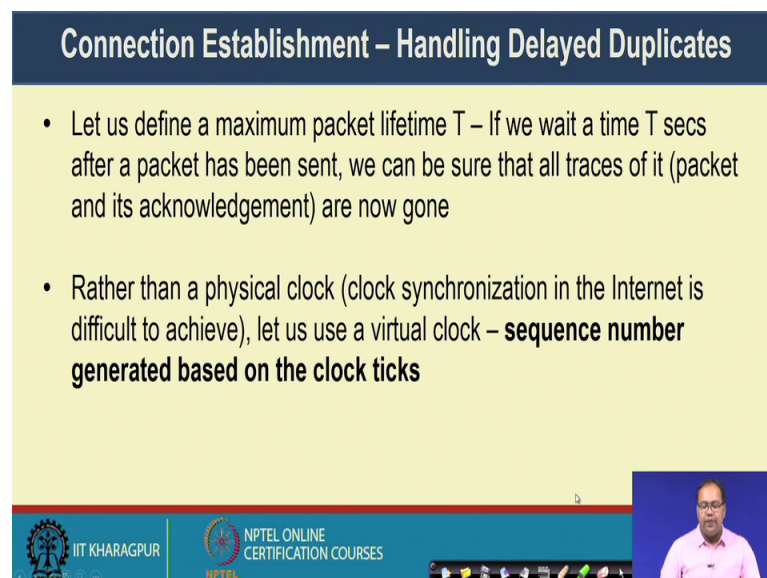
When over entire design challenge here is that, we need to guarantee not only that a packet is dead, but all acknowledgement of it are also dead. So, this is an interesting requirement, because whenever you are sending a connection request message it may happen that from the server side and here is the client side say from the client side, you have sent a connection request message and then the client got crashed and it has restarted again say it has restarted again at this point, now here it receives the reply message.

Now, if it replies the reply message and just before sending, the reply message if it has sent another connection request. Then by looking into this reply the client will be in a dilemma up whether this reply is the reply corresponds to the old request or it is the reply corresponds to this new request that it has just sent out because remember this. So, all though for the explaining purpose I am marking it has blue and brown, but the client cannot see it as a blue or brown.

So, the client just looks into that it is a reply to the connection request message that it has already sent out and it has got a reply. So, it is in a dilemma or it will not be able to correctly decode whether that reply is the delayed duplicate or because of this crash failure the reply of the earlier connection request that it has said. So, we need to design mechanism to prevent this kind of things so that the client actually be able to differentiate between this blue and brown.

And it can find out that well the reply message that it has received it is the reply corresponds to the blue request and not the brown request and it can correctly drop that particular reply message. So, we need to guarantee that not only a packet is dead, but all acknowledgement to that packets are also dead.

(Refer Slide Time: 25:06)



Connection Establishment – Handling Delayed Duplicates

- Let us define a maximum packet lifetime T – If we wait a time T secs after a packet has been sent, we can be sure that all traces of it (packet and its acknowledgement) are now gone
- Rather than a physical clock (clock synchronization in the Internet is difficult to achieve), let us use a virtual clock – **sequence number generated based on the clock ticks**

The slide is part of an NPTEL presentation from IIT Kharagpur. It includes a video inset of a speaker in the bottom right corner and logos for IIT Kharagpur and NPTEL Online Certification Courses at the bottom.

So, let us see that how we can do this or how we can handle the delayed duplicates during the case of connection establishment. So, we define the maximum packet lifetime t . And we make it sure that if we wait for this T duration, then if you wait for this T

duration then, you can be sure that all traces of it; that means, the packet and also its acknowledgement have now gone from the network. So, all the packets and all the traces of it are acknowledged and accepted.

Now, to ensure that in case of a transport layer protocol which is also utilized in the concept of TCP. So, rather than using a physical clock because the problem of having a physical clock is that you require clock synchronization which is difficult to achieve in the internet scale, we use the concept of virtual clock. So, what is this virtual clock? This virtual clock is a sequence number field which is generated based on the clock ticks. So, it is just like that every individual packet that you are sending out, that individual packet will contain a sequence number. And by looking into the sequence number you will become sure whether that particular packet was the intended packet or not.

So, the question comes: says that how will you design that sequence number or whether there is still a problem even if you design a sequence number mechanism.

(Refer Slide Time: 26:39)

Connection Establishment – Handling Delayed Duplicates

- Label segments with sequence numbers that will not be reused within T secs.
- The period T and the rate of packets per second determine the size of the sequence number – **at most one packet with a given sequence number may be outstanding at any given time**

The slide footer includes the IIT Kharagpur logo, the NPTEL Online Certification Courses logo, and a small video inset of a man in a pink shirt speaking.

So, here is the broad idea that you label every segment with a sequence number, and that particular sequence number will not be reused within that T second duration. So, what we say is that within that T second duration every segment or every packet that I have sent into the network, it will die off as well as all traces of that packet. That means if there is certain acknowledgement for that packet, they will also get die off. So, with this particular principle you can say that if you are not going to reuse that

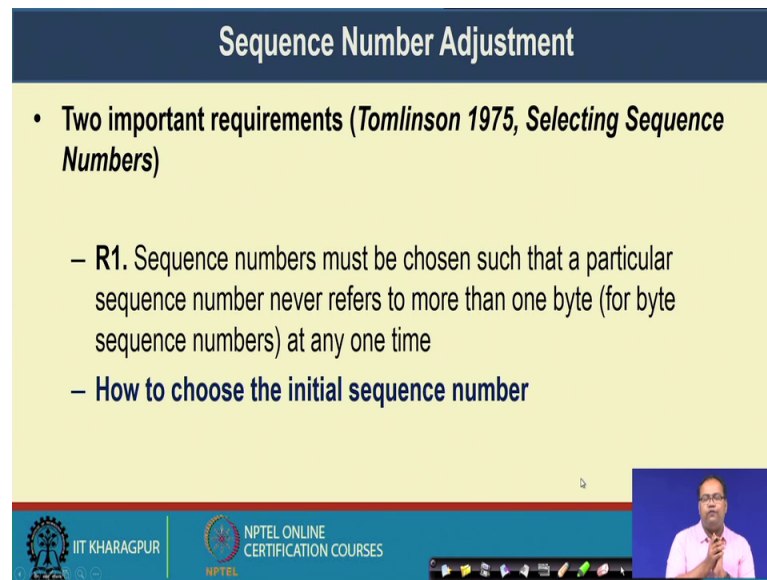
sequence number, within that T second of duration you will be able to ensure that at any time, there would be only a single instance of a packet with a unique sequence number.

So, just giving you an one example say you have transfer the packet of say sequence number 125, sequence number 125 and you say T equal to 1 minute; that means, you are trying to ensure that once you have transmitted packet with say sequence number 125 within this 1 minute duration, this particular sequence number 125 is not going to be reused. If you can ensure that then you know that after 1 minute duration, the packet that you have send to it sequence number 125 that is going to die off from the network.

So, the packet will be there in the network for 1 minute and within that 1 minute duration, if you are not sending any under packet with the same sequence number the same sequence number 125. Then you will be sure that well no cases of this packet no other cases or the duplicate cases of the packets will be there in your network. So, that way you will be able to ensure that whenever the other end will receive a packet with this sequence number 125 that is the only packet that is a towards in the network or not a delayed duplicate of that particular packet. So, this period T and the rate of packets per second determined the size of the sequence number.

So, we want to ensure that atmost one packet with a given sequence number maybe outstanding at any given time. So, it is just like that once you have sent a packet with a sequence number 125 within that T second duration or within that T duration, you do not send any other packet with the same sequence number. So, only that packet with the sequence number 125 is outstanding in the network within that particular duration.

(Refer Slide Time: 29:31)



The slide is titled "Sequence Number Adjustment" in a dark blue header. The main content area is yellow and contains two bullet points. The first bullet point is "Two important requirements (Tomlinson 1975, Selecting Sequence Numbers)". The second bullet point is "R1. Sequence numbers must be chosen such that a particular sequence number never refers to more than one byte (for byte sequence numbers) at any one time". Below this is a sub-bullet point "How to choose the initial sequence number". The slide footer is blue and contains the IIT Kharagpur logo, the NPTEL Online Certification Courses logo, and a small video inset of a speaker.

Sequence Number Adjustment

- Two important requirements (Tomlinson 1975, Selecting Sequence Numbers)
 - R1. Sequence numbers must be chosen such that a particular sequence number never refers to more than one byte (for byte sequence numbers) at any one time
 - How to choose the initial sequence number

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, here we have 2 important requirements that we need to ensure. So, this 2 requirement was published by Tomlinson in 1975 in a part breaking work titled selecting sequence numbers. So, the first requirement is that the sequence number they must be chosen such that a particular sequence number refer never refers to more than 1 byte. So, if you are using byte sequence numbers. So, byte sequence number means that for every individual byte that you are sending in the network they has a sequence numbers.

So, that TCP type of protocol it uses byte sequence number rather than the packet sequence numbers. So, in case of a packet sequence number for every individual packet that you are transferring in the network, you put one sequence number for the packet, for the byte sequence number every individual byte that you are transferring in the network you put one sequence number for that. So, the byte sequence number is something like this like if your packet has some 100 byte data.

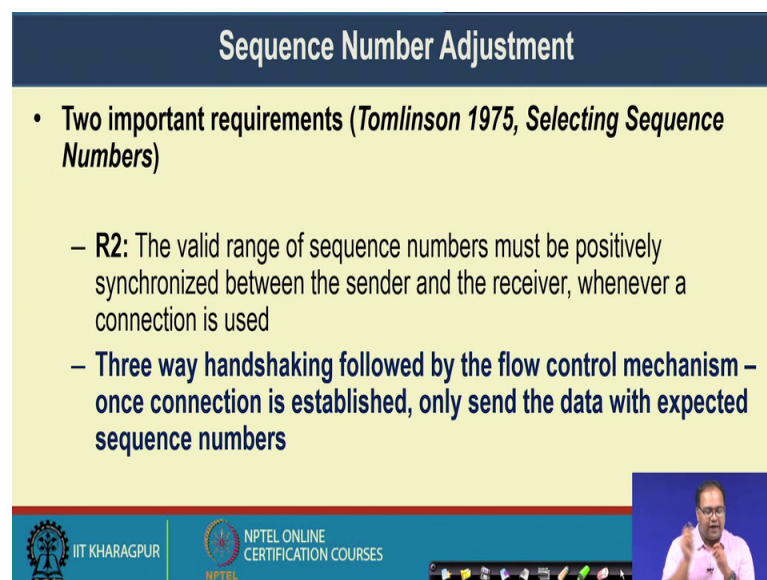
So, the packet has 100 byte data so, in the header field you have 2 different field. One is this sequence number and other is the length so, the lengths says that you have 100 byte data the sequence number field is a 500; that means, in this particular packet you have data from 500 bytes to 600 bytes 501 bytes to 600 bytes. So, you have total 100 bytes of data.

So, that way you can use the byte sequence numbering 2 individually identify every bytes in the networks. So, that would be useful later on you see for ensuring segment

wise delivery on top of a transport layer protocol. So, the requirement here is that every sequence number that you are sending to the network it indicates to only a single byte not more than 1 bytes so, there should not be more than 1 bytes in the network for the same source destination pairs which are reference by a single sequence number.

Now, in this case the challenge comes that how will you choose the initial sequence number. The initial sequence number is required during the connection establishment phase, when you are trying to send data to a remote host. So, that was the first requirement you will see that how you can choose the initial sequence number during the connection establishment phase.

(Refer Slide Time: 32:05)



The slide is titled "Sequence Number Adjustment" in a dark blue header. The main content area is yellow and contains two bullet points. The first bullet point is "Two important requirements (Tomlinson 1975, Selecting Sequence Numbers)". The second bullet point is "R2: The valid range of sequence numbers must be positively synchronized between the sender and the receiver, whenever a connection is used". The third bullet point is "Three way handshaking followed by the flow control mechanism – once connection is established, only send the data with expected sequence numbers". The slide footer is blue and contains the IIT Kharagpur logo, the NPTEL Online Certification Courses logo, and a small video inset of a speaker.

Sequence Number Adjustment

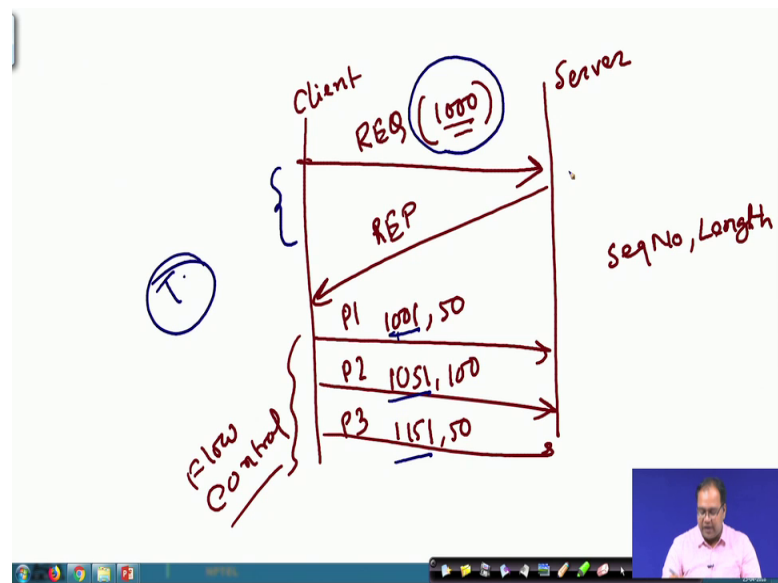
- **Two important requirements (Tomlinson 1975, Selecting Sequence Numbers)**
 - **R2:** The valid range of sequence numbers must be positively synchronized between the sender and the receiver, whenever a connection is used
 - **Three way handshaking followed by the flow control mechanism – once connection is established, only send the data with expected sequence numbers**

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

And the second requirement is that the valid range of sequence number must be positively synchronized between the sender and the receiver, whenever a connection is being used. So, this means that whenever you have set up is initial sequence number then, all the subsequent bytes will follow that sequence number. So, this is basically ensured by the flow control algorithms.

So, later on will see the different types of flow control algorithms, which actually ensures that once the sender and the receiver or the client and the server has agreed upon the initial sequence numbers then the flow control algorithm ensure that well the packets or the bytes that you are going to transfer, it follows that sequence of the sequence number.

(Refer Slide Time: 32:58)



So, the one example can be something like this say you have a client and you have a server. Now the client sends request message with say initial sequence number as 1000 and the server sends a reply mentioning that it accepts the initial sequence number as thousand. Now once this connection establishment is being done, then all the subsequent packets that is being sent by the client it follows this sequence number space.

So, the first packet say it will start from 1000 to 1 and it has the length of 50 bytes. So, this things I am writing in the form of sequence number comma length so; that means, the first packets starts from 1000 2 1 and it has a length of 50. The second packets starts from then 1051 and it can have a length of 100 then the third packet starts from 1 1 5 1 and it can salient of another 50.

So, this particular thing the sequence number is that at what sequence the packets will we transferd that is handled by the flow control algorithm. So, later on will see that how flow control algorithm actually ensures that. So, this particular mechanism we call it at between the client and the server between the 2 ends you should have a positive synchronization for ensuring that every individual packets are having following the sequence number, which have been established during this initial handshaking phase and the sequence number in follows that particular principle.

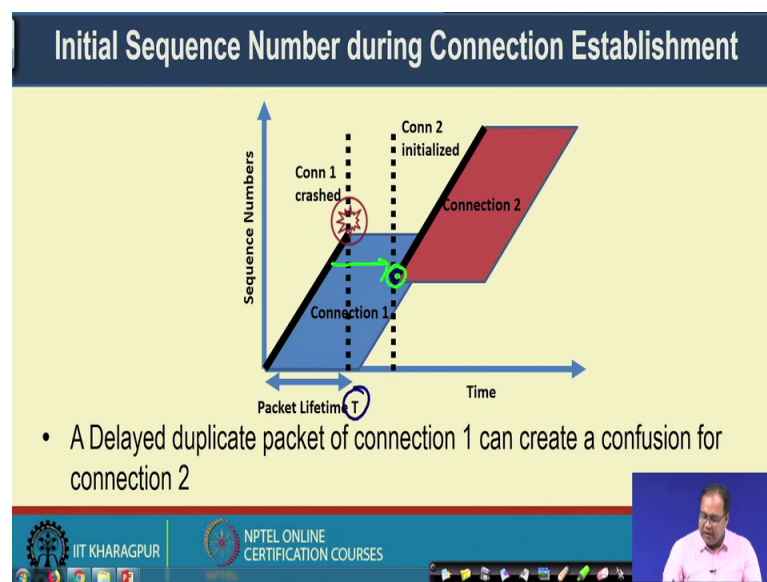
Now, here you will see that once this initial handshaking is done, the problem is gone the problem will be taken care of by the flow control algorithm, but the problem is the first

requirement which was there, that how will you choose this initial sequence number. Because for this subsequent packets say this is packet one this is packet 2 this is packet 3, for the subsequent packets you have this referencing the reference of the sequence number that which particular sequence number you are going to use based on what sequence number has already been utilized.

So, this individual sequence number like 1000s 1051 11 51 they are known once this initial hand shaking is done, but this initial sequence number it is unknown. So, that need to be establish and during this establishment of the initial sequence number you need to ensure that whichever initial sequence number you are going to use, that is not going to be reuse to within certain duration of t.

So, that time bound need to be there and within that time duration that initial sequence number is not going to be reused such that the server it can differentiate between a correctly send connection request and the delayed duplicate of it. So, that is the broad requirement that we have in the context of connection establishment.

(Refer Slide Time: 36:16)

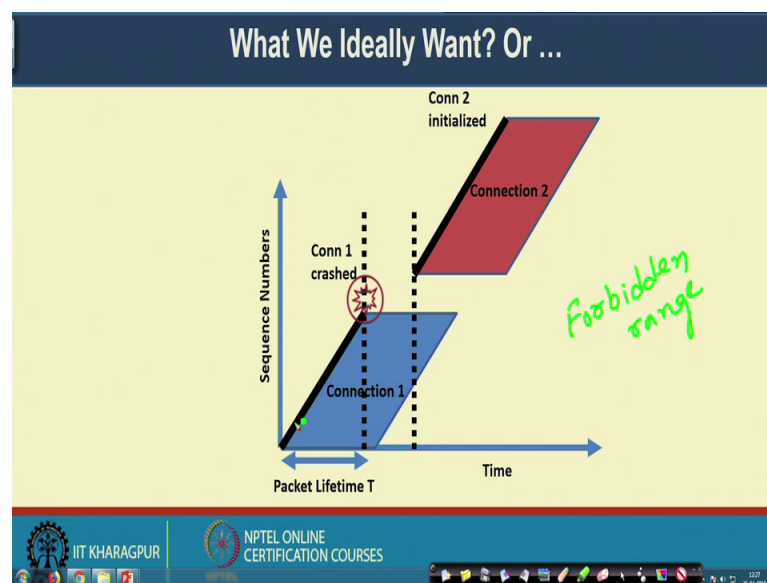


Well this is the problem that we have like once a particular machine it is trying to send the data it has chosen one initial sequence number, and it is transferring the data on top of the network and we have a packet lifetime T. And; that means, every by that you are sending using this sequence number filled, that will be there in the network for this time duration T.

Now if this connection get crashed and if you are initiating another connection with this initial sequence number say with this initial sequence number then the problem is that you can see that here you have 2 different packets you may have 2 different packets, which are there in the network one is the old packets from the connection 1 which was still there in the network and the new packet from connection tool.

So, there can be a confusion; so, we want to avoid this kind of confusion here, that we one that well the connection to should not initiate from this point. Rather a connection to will either initiate from this point. So, you wait for sufficient amount of duration, and then initiate the new connection with a new sequence number. So, that you can become sure that this connection 1 and a connection 2 there sequence number fill does not get overlap does not get overlap.

(Refer Slide Time: 37:51)



Or the second thing is that you use the sequence number which is high enough from the sequence number fill that you have used for the connection 1. During that time, you also be able to ensure that the sequence number zone of connection 1 and connection 2 they does not get overlap and there is no confusion in the sequence number. So, that is our requirement.

So, you want to either wait for a duration so, that we make ensure that all the previous bytes with the old sequence number that are gone out of the network or you use a initial sequence number, which is high enough compared to the previous sequence number that

has been utilized for this connection establishment. So, that the connection zone of 2 nodes they does not get to each other. So, here in this diagram this, particular zone this blue zone or here this red zone we call is a forbidden range. So, we call it as a forbidden range ok. Because once one sequence number is being used you should not reuse the sequence number any more.

So, in the next class we will look into the details about how you can design a mechanism for selecting the initial sequence number so, that you can avoid the overlapping of the forbidden zones for 2 different connection. So, see you all in the next class.

Thank you.