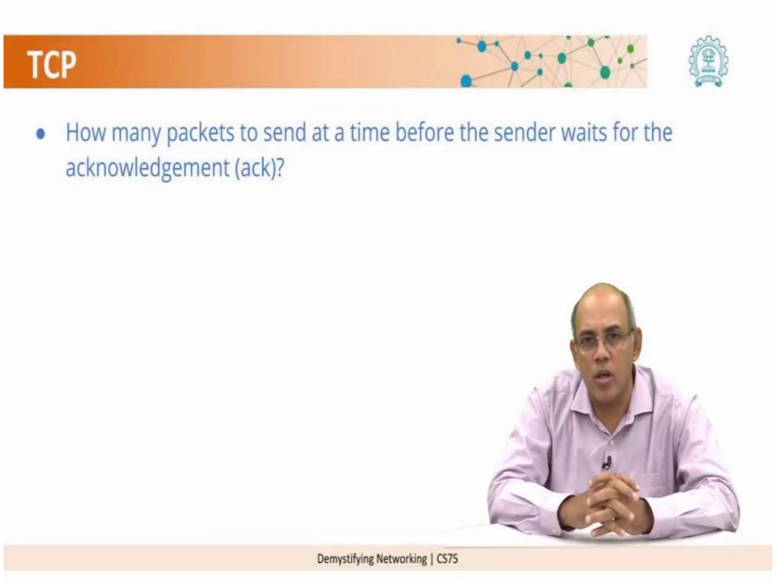
Demystifying Networking

Prof. Sridhar Iyer
Department of Computer Science and Engineering
Indian Institute of Technology, Bombay

Lecture - 57
Introduction to TCP

The next question that we want to address is, how many such packets should we send at a time.

(Refer Slide Time: 00:07)



TCP

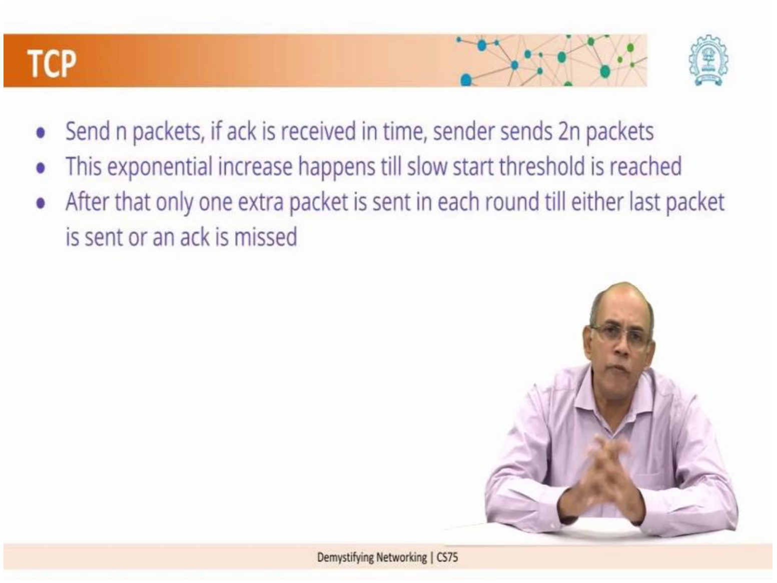
- How many packets to send at a time before the sender waits for the acknowledgement (ack)?

Demystifying Networking | CS75

So, let us take a pause here and consider this question ok. There is a sender at one end of the network, there is the receiver at the other end of the network, now how many packets should the sender send at a time before the sender waits for the acknowledgement to come back. At one extreme you can send one packet wait for the acknowledgment then send the next packet wait for the acknowledgement. At another extreme you might be able to send hundreds of packets or all the packets at the same time and maybe some of them will reach some of them will not reach wait for the acknowledgement and so on.

So, what is a systematic way of slowly increasing the number of packets that can be sent along the pipe? Take a moment to think about this and when you are ready then we can move on after the reflection spot ok.

(Refer Slide Time: 01:05)



TCP

- Send n packets, if ack is received in time, sender sends 2n packets
- This exponential increase happens till slow start threshold is reached
- After that only one extra packet is sent in each round till either last packet is sent or an ack is missed

Demystifying Networking | CS75

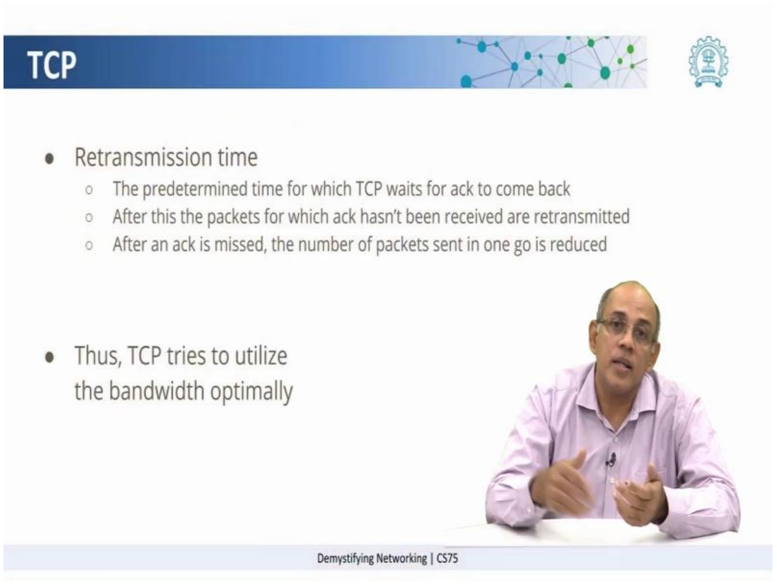
So, some of you may have thought that it would be useful to send a number of packets and if the acknowledgements do not come back then we reduce the number of packets, others may have thought that we should be a little conservative we should send a few packets first and if the acknowledgements reach, then we can send a few more packets and so on.

Both these ideas are correct actually what TCP does is a mix of both of them. There are many variants of TCP. In one of the variants essentially what it does is it starts with one packet, if the acknowledgment for that packet is received back in time the next round it sends twice the number of packets. Once again if the acknowledgement is received back in time, in the next round it sends twice the number of packets. So, in that sense it grows exponentially up to a certain point.

Now; obviously, it cannot keep on growing exponentially forever. So, there is a threshold. So, this threshold is in technical terms called the slow start threshold and so, up to that number of packets, it grows exponentially and then after that it sends one extra packet in each cycle. So, for example, if in the first round you send 1, in the next round you send 2, in the next round you send 4, in the next round you send 8 and if 8 is the threshold, after 8 in the next round you send 9, in the next round you send 10 in the next round you send 12. And you go on with this process till either you have finished the transmission or at some point some acknowledgement does not come back within the expected time.

How does TCP know how much time to wait for the acknowledgment?

(Refer Slide Time: 02:49)



TCP

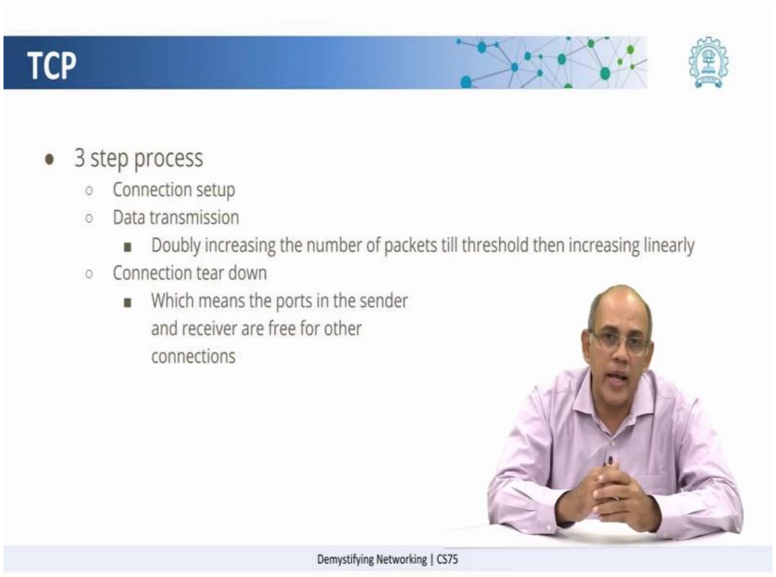
- Retransmission time
  - The predetermined time for which TCP waits for ack to come back
  - After this the packets for which ack hasn't been received are retransmitted
  - After an ack is missed, the number of packets sent in one go is reduced
- Thus, TCP tries to utilize the bandwidth optimally

Demystifying Networking | CS75

That again is a new concept which is called the retransmission time out. So, at this point we can just think of it as a predetermined time that TCP waits for the acknowledgement to come back. If the acknowledgement does not come back in time, then once again it starts, it drops the values and it starts again from a smaller value of number of packets to be sent.

So, in this manner what TCP tries to do is, it tries to utilize the network bandwidth optimally in sending the packets from the sender to the receiver.

(Refer Slide Time: 03:29)



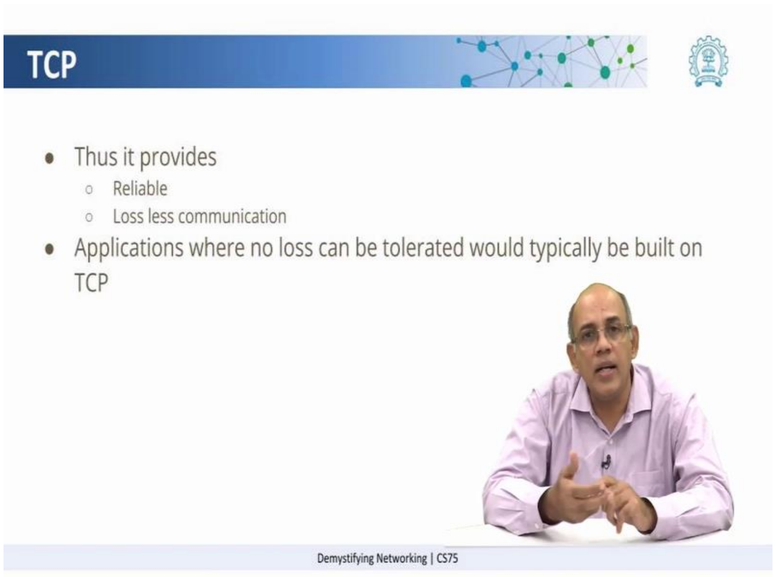
TCP

- 3 step process
  - Connection setup
  - Data transmission
    - Doubly increasing the number of packets till threshold then increasing linearly
  - Connection tear down
    - Which means the ports in the sender and receiver are free for other connections

Demystifying Networking | CS75

So, to summarize we have the three steps in the entire process; one is the connection setup, where there is a connection which is established between the sender and the receiver, then there is the data transmission; within the data transmission we have this notion of increasing the number of packets by doubling them up to a certain threshold, and then increasing them linearly till the entire transmission is complete. And then after the transmission is complete, there is the idea of tearing down the connection or terminating the connection, so that the sender and receiver, those ports on the sender and the receiver are now free to be used for other connections.

(Refer Slide Time: 04:13)



TCP

- Thus it provides
  - Reliable
  - Loss less communication
- Applications where no loss can be tolerated would typically be built on TCP

Demystifying Networking | CS75

So, this is the way in which TCP ensures that packets are delivered reliably and without errors or without loss. So there are a number of applications that use TCP as the underlying thing, in any application where the loss of some information would be crucial would typically be built on TCP.