Broadband Networks

Prof. Karandikar

Electrical Engineering Department

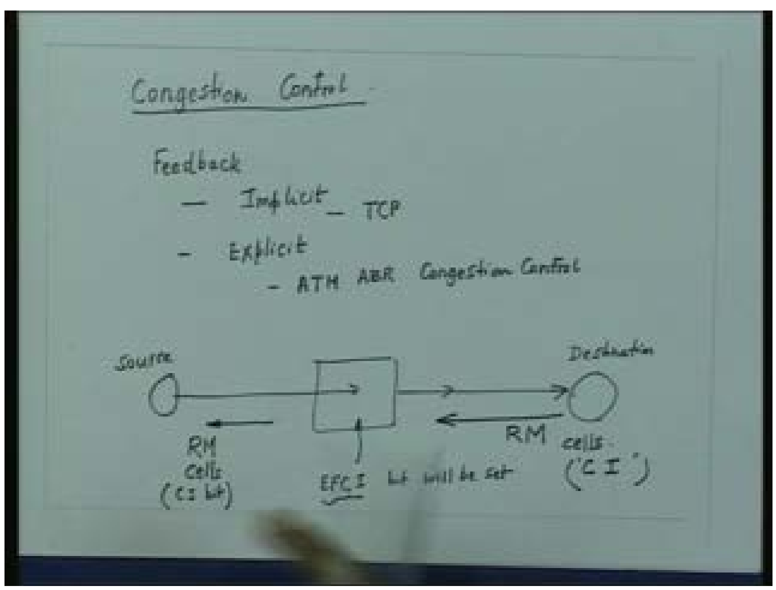
Indian Institute of Technology, Bombay

Lecture - 16

So, till now we have considered the control of sources which have the rho sigma regulator at its input. That means the sources are rho sigma regulated. We will now see how can we control the best effort traffic in an internet and essentially the control of best effort traffic in internet is achieved by some kind of congestion control techniques. And, there are various congestion control techniques which are used in today's networks and we will see one of the most popularly used a congestion control technique that is the TCP's congestion control techniques.

The TCP is a transmission control protocols, we know about its elementary features but what we will discuss today is about the congestion control aspect of the TCP.

(Refer Slide Time: 2:25)



Now, let me just point out that the congestion control techniques if you classify the congestion control technique; now the congestion control techniques can be classified based upon the kind of feedback that you receive and this feedback can be either implicit or it can be explicit. By implicit feedback we mean that an indication about the congestions in the network is conveyed to the congestion control mechanism in an implicit manner. One of the examples of an implicit feedback use is the TCP's congestion control technique.

Now, in TCP the source comes to know of the congestion in the network if the acknowledgment from the receiver does not reach the source within a certain time of interval. So, what happens is that when the source transmit the packets, it expects an acknowledgment from the receiver within a certain time of period and if the acknowledgment does not reach within the time out period, then the source infers that the packet must have been lost and that is the reason the acknowledgment is not coming and the loss of packet signal is taken as an indication of congestion in the network.

Now, we will see later that this may be a reasonable assumption to make in the wired packet switched networks. But if the bottleneck link happens to be the wireless networks, then it may also be possible that the packet loss might have occurred on the link due to the time varying nature of the channel also. So, that is also might be possible that a packet might have been lost due to the channel fading. Essentially, the packets may be in error due to the channel fading and that is why the packet might have been discarded by the node.

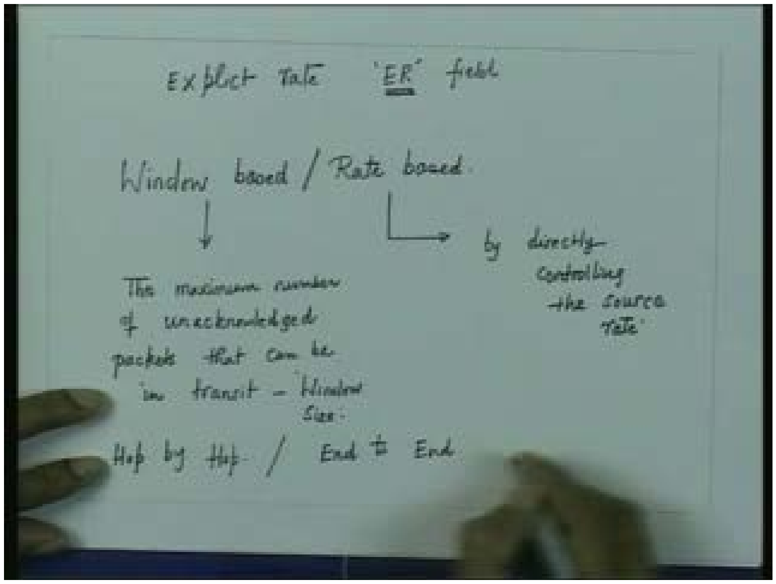
So, the loss of packet can also occur because of the channel fading as well. But as far this discussion is concerned; we will assume that a packet is lost because of congestion in the networks. However, in another technique, the feedback from the network about the congestion conditions can be explicit also. The congestion control techniques which relies upon the explicit feedback is ATM's ABR congestion control techniques. So, the implicit feedback is used by the TCP's congestion control technique, the explicit feedback is used by the ATM ABR congestion control.

We have already discussed what is the ABR traffic in our previous lectures. We said that ABR traffic stands for available bit rate and this traffic is not given any quality of service guarantees. But instead it uses whatever is the available bandwidth in the network. Now, the ABR's congestion control mechanism basically uses an EFCI bit. Now, when the ATM cells pass through the switch and if a particular switch let us say is congested; so, the scenario is something like this that you may have a node in between here and your ATM cells are passing like this, so this is an intermediate node and this is a source and let us say that this is the destination and these are the intermediate nodes.

So, when the ATM cells pass through it and if this particular switch is congested, then this ATM cell will switch will set the EFCI explicit forward congestion indication bit, it will be set and when this ATM cell reaches the destination and when the destination finds that the EFCI bit has been sent and the destination sends the resource management cells which is called as RM cells; the resource management cells will have their congestion indication bits set - CI bit set. So, when the resource management cells go to the source, I mean when they are received at the source and if they have the CI bit set; then the source comes to know that congestion has occurred at a particular switch and then the source will reduce its rate of transmissions.

Now, in the explicit feedback in the ATM world, explicit rate can also be set by the congested switch.

(Refer Slide Time: 7:54)



So, what may happen is that in the explicit feedback congestion control, explicit rate can also be set. Explicit rate, there may be ER field also. Now, as you have seen in this case what was happened is that if the ATM switch is congested, it will set the EFCI bit which is just an indication to the destination that along this path some switch is congested. However, if there is an ER field is there, then this particular switch can also set what is the rate that it is expecting.

Now, this ATM cell may already be having some rate field inside. Now, if this rate happens to be larger than what the switch expects the source to transmit; then it can reduce its rate and finally when this field reaches the destination, the destination will know that what is bottleneck rate and it may inform to the source through the resource management cells.

So, this is like the feedback. As I said, the congestion control mechanisms can be classified based upon the feedback they received about the congestion conditions in the network and this feedback can be implicit in the form of a in the form of packet loss being an a indication of congestions or this feedback can be explicit where the intermediate nodes are explicitly telling the congestion control mechanism that a congestion has occurred at a particular node.

The congestion control mechanisms can also be classified based upon how they are controlling the congestions. Accordingly, we can have a window based mechanism or a rate based mechanism. So, the other technique is that we can have window based congestion control mechanisms or rate based congestion control mechanisms. In window based congestion control mechanisms, there is a particular window size let us say w. What does it mean is a source can transmit as many as w packets without waiting for an acknowledgment.

So therefore, the window size determines the maximum number of unacknowledged packets that can be there in transit that can be there on the pipe. So, the window size determines the maximum number of maximum number of unacknowledged packets that can be in transit is called as the maximum window size.

So, by limiting this maximum window size that is by limiting the number of unacknowledged packets that can be there in the transit, a window based congestion control mechanism can control the congestions. In the rate based congestion control mechanisms, you are directly controlling the source rate. So, a rate based congestion control mechanism operates by directly controlling the source rate and a window based congestion control mechanism operates by limiting the maximum amount of unacknowledged data that can be there in the transit.

For example; TCP's congestion control mechanisms. TCP's congestion control mechanisms operates by limiting the maximum number of unacknowledged packets that can be there in the transit. So therefore, TCP is a window based congestion control mechanisms. We can also classify the congestion control mechanism based on whether they are hop by hop or whether they are end to end. By hop by hop we mean that the congestion control is exercised on node to node basis. On each intermediate node, we are exercising the congestion control mechanisms.

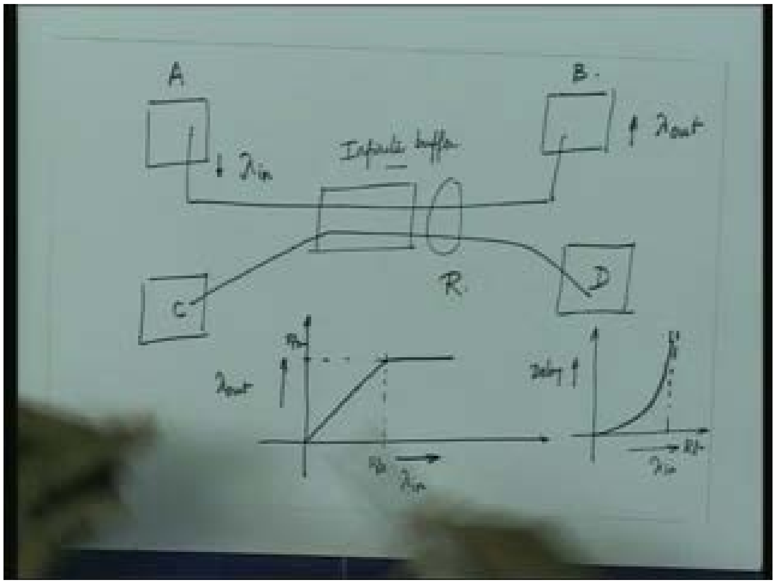
End to end: in an end to end congestion control mechanisms, the congestion control mechanism operates between the source and destination pairs on an end to end basis. The intermediate nodes do not participate in the congestion control of a particular traffic flow of a particular traffic flow that exists between a particular source and a particular destination. An end to end congestion control mechanisms will operate between the source and the destination for a particular traffic flow. So, that congestion control mechanism can be hop by hop or it can be end to end.

So, have you all seen that the congestion control mechanisms that exist today to control the quality of best effort traffic in the internet; we can either classify them based on the feedback these mechanisms receive, based upon whether they are window based or rate based that means how they are controlling the congestions, how they are observing the conditions about the networks and third one how they operate - whether they operate on a hop by hop basis or whether they operate on a rate by rate basis.

Now, before we go into the specifics of the congestion control mechanisms, what I would like to explain is one should view the congestion control mechanisms as a resource allocation mechanism for the best effort traffic. Now, note that for the best effort traffic in statistical multiplexers or in a packet switch node; we are not giving any explicit quality of service guarantees, we are not giving any bandwidth guarantees, we are not giving any delay guarantees we are not giving any delay guarantees.

However, the packet switch network must ensure that all the flows share the output link bandwidth of the node first of all in a fair manner a and b. whenever since there is no admission control and since there are no guarantees, the sources can pump data into the networks; there must be a mechanism of conveying to the sources if the congestion condition is developed in the network. There must be a mechanism of conveying the sources and the sources must reduce their rate of transmissions. So, to appreciate that let us see a very ideal well scenario that why do we need a congestion control mechanism.

(Refer Slide Time: 15:57)



Let us assume that we have one node here and there is a source A and the destination B here and there is another pair C and there is a destination D. So, A is talking to B and C is flow exist from D and this is on the same output link over here. There may be a intermediate nodes also and this output link capacity is let us say R and the input rate here, the offered rate is let us say is lambda $_{in}$ and the output rate that is effective throughput has seen by the B may be let us say lambda $_{out}$.

Now, let us say that this node has an infinite buffer. Now, if this node, this intermediate node have an infinite buffer; now what happens is now let us consider the traffic flow between A and B. Consider the traffic flow between A and B, what happens is that if I increase the lambda $_{in}$, my lambda $_{out}$ is going increase. So, this is I am plotting lambda $_{in}$ verses lambda $_{out}$. So, if I increase lambda $_{in}$, my lambda out going to increase because the link has still lots of capacity and when my lambda $_{in}$ becomes R by 2, my lambda $_{out}$ will become R by 2. Assume that link has capacity of R.

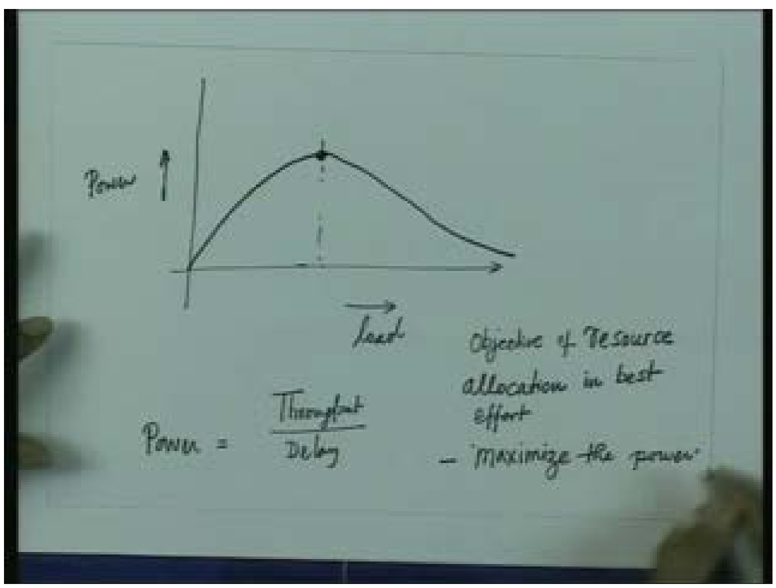
Now, after that if I try to increase this lambda $_{in}$, then my lambda $_{out}$ that is the effective throughput that this output is likely to be only R by 2. The reason being now is that this link R needs to be shared equally between the 2 flows that exists: between the flow of A and B and between the flow of C and D. So therefore, R by 2 throughput is achieved by the AB flow and again R by 2 throughput is achieved by the flow of CD and we are assuming here that this flow CD is also experiencing the same congestion control.

Now, this is assumed to be an infinite buffer. So, as we keep on increasing the lambda $_{in}$; what is going to effect? If you see on the delay characteristics, so if this was a delay and this is my lambda $_{in}$; so as I increase lambda $_{in}$, my delay slowly keeps on increasing and when I reach R by 2 here, my delay tends to infinity.

So, what happens is that if I keep on increasing the offered load or the offered load into the system, my throughput will increase first but at a certain point, my delay will start increasing and if there is an infinite I am assuming here in very ideal scenario where there is an infinite buffer and there may not be any packet loss. Now, if this buffer… so even though there is no packet loss even though there is a no packet loss, we require a mechanism to now control the source rate. We should not be transmitting or pumping too many packets into our node. Otherwise, the delay will increase indefinitely. Now, this is an ideal scenario where we assumed it to be infinite buffer.

But suppose, this buffer was finite and therefore what will happen is that there will be a packet loss. The moment there is… assume again ideal scenario where the source knows exactly that a packet loss has occurred. Now, when a packet loss has occurred, the source retransmits the packet. So therefore, again if the effective throughput goes down because now the data also contains not only the original data but it also has the retransmitted packets; still like an ideal scenario we have considered. But in a network of where there are multiple hops and where there are multiple flows are traversing a particular node, it has been found that we can define some kind of a which is called as the power.

(Refer Slide Time: 20:42)



So, what we can say is that that we can plot offered load verses power and this power is nothing. The power of a flow is defined as the throughput upon delay. So, what happens is that if I keep on increasing the load, my power keeps on increasing and I reach a point and after that if I increase the load, my power actually decreases and eventually the power may become 0.

So, at this point essentially what is happening is that if I keep on increasing the offered load, my throughput increases dramatically, throughput increases at a faster rate and when I reach this point; after this point my delay start increasing at a faster rate and I may reach a point where the delay becomes infinite, the delay becomes very large or else the throughput may become 0 because of the packet losses and therefore actually the power may go to 0.

So now, one should view a congestion control mechanism as a resource allocation mechanism. We would like to operate our source, we would like to have the source operating point to be at this point to be at this point, where my power is maximized. So, what I am trying to say is that the objective of a resource allocation objective of resource allocation in best effort network should be to maximize this power. Our congestion control mechanism should operate in such a manner that it maximizes this power. It makes this source operate at this point.

Now, most congestion control techniques therefore try to achieve somewhere, some kind of, somewhere this kind of objectives. So, that is one thing that that should be the objective of a resource allocation mechanisms that it must maximize the it must maximize the power of a particular connection or of a particular flow between the source and a destination. Another objective that it should have, a resource allocations in the best effort network should have; it should be fair.

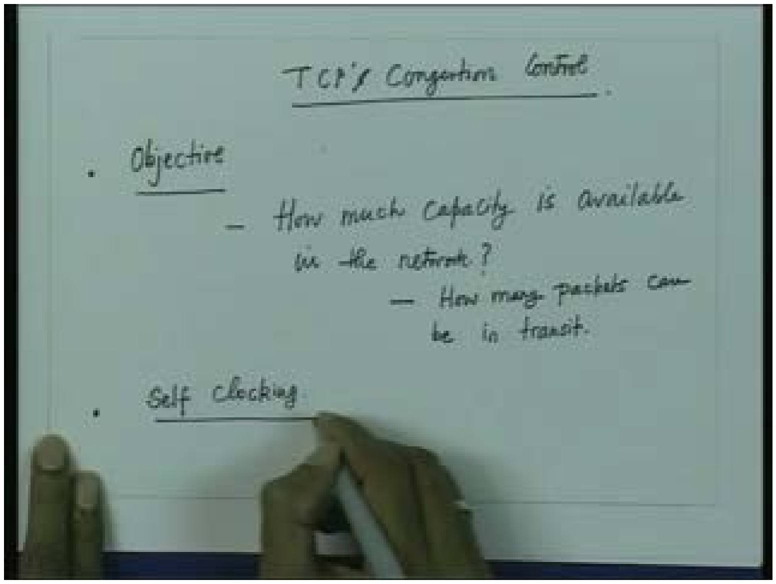
We have already considered the definitions of the maxmin fairness and the fairness can be achieved by using some kind of a combination of a buffer management combination of a buffer management and a packet scheduling algorithms. So essentially now, one can see, even though we are saying that in a statistical multiplexers of packet switch networks; we are not offering any quality of service guarantees. Even then, we would try to give some quality by using an effective congestion control mechanisms and effective resource allocation mechanisms in combination with the packet scheduling and the buffer management algorithms.

Packet scheduling algorithms, we have already seen to provide fairness. We had already seen how packet scheduling algorithms can also give quality of service guarantees in terms of delay bounds. We have also seen how they can guarantee a particular delay bound by enforcing the sources to be rho sigma regulated. But they can also give you fairness like a weighted fair queuing and other kinds of algorithms can also give you fairness.

So, fairness is one issue. We will later on in our next lecture we will discuss about the buffer management schemes - that we need to discuss. But right now we are discussing, how we can have an effective congestion control mechanisms.
\
So, we will first discuss TCP’s congestion control mechanisms. We will then briefly discuss ATM’s ABR congestion control well mechanisms and then we would also see what kind of rate based congestion control mechanisms are used for the multimedia streaming over internet and so on which are using udp as the transport protocol. So, that we will see in our subsequent lectures.

Now, let us today see, how does the TCP’s a congestion control mechanisms operates and things like this.

(Refer Slide Time: 25:40)

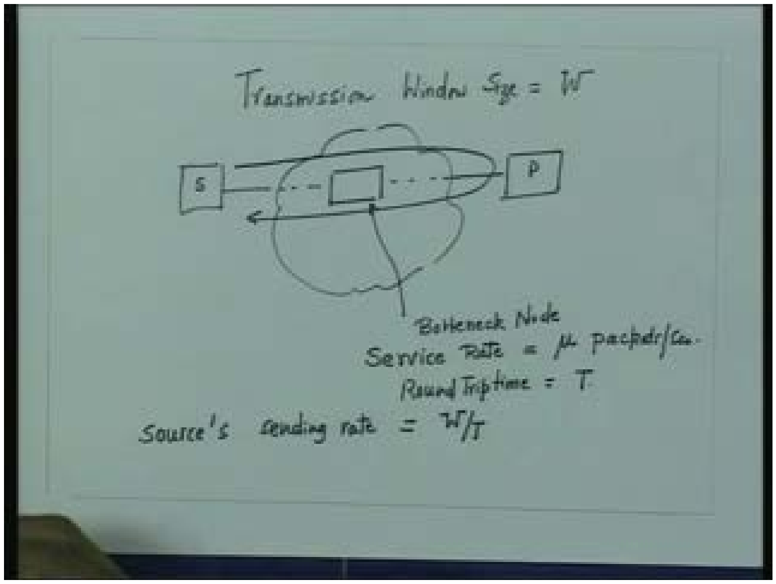


So, what we are saying is the TCP's congestion control mechanism. Now, the TCP's congestion control mechanism really works with an objective. It tries to determine how much capacity is available in the network. So, the TCP's congestion control mechanisms operates by saying how much capacity is available in the packet in the network and therefore how many packets can be injected into the network and the technique is essentially based upon the window based congestion control mechanisms.

So, by determining how much capacity is available in the networks; you can determine how many packets can be pushed or can be in transit. So, that is the objective and the second thing is that the TCP's congestion control mechanism is self clocking. By self clocking we mean that whenever an acknowledgment arrives from the destination, its signals that a packet can be injected into the network. So, arrival of an acknowledgment will signal that a packet can be injected into the network. So therefore, the mechanism is self clocking. The arrival of an acknowledgment will indicate that packet can be safely injected into the network. We call this to be self clocking.

Now, thus as I said that the TCP's congestion control mechanism operates by limiting the maximum number of unacknowledged packets that can be there in the transit but it limits this. So essentially, TCP's congestion control algorithm dynamically adjusts the window size. TCP's congestion control algorithm will dynamically adjust the window size. So, to understand the relationship between the window size and a particular bottleneck source rate, let me just explain this how that relationship exists.

(Refer Slide Time: 28:53)

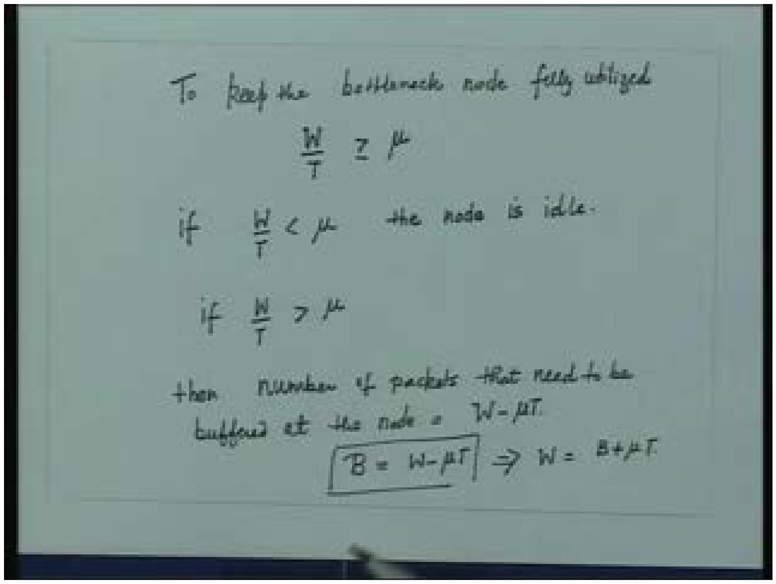


Let us say that the transmission window size is W. So, let me just explain that will give you an idea that what is the relationship between the TCP's congestion control mechanisms and the window size. So, let us say that let transmission window size be W. Now, let us say that a particular node is a bottleneck node. So what happens is that this is a source and you send in a network and this is a destination and this particular node is the congested node.

So, congestion has occurred. So, we call this to be the bottleneck node and let us say that the bottleneck's node service rate is mu packets per second and let us assume that round trip time from the source to the destination is T. So, this round trip time is T seconds. So, since the round trip time is T seconds, so what happens is that if a packet is sent from the source to destination and the acknowledgment coming back that much time will be actually the capital T.

So therefore, the source sending rate which is actually equal to W by T. The window size is W, the round trip time is T; the source sending rate will be W by T. Now, in order to keep the bottleneck link fully utilized, you want to keep the link fully utilized; obviously, W by T has to be greater than or equal to mu.

(Refer Slide Time: 31:20)



So, in order to keep the bottleneck link fully utilized bottleneck node fully utilized; we need W by T to be greater than mu. Now obviously, if W by T is less than mu, then node is actually ideal. If the source sending rate is less than the bottleneck service rate, obviously the node is ideal. If the source sending rate happens to be greater than the bottleneck service rate, then the bottleneck link is fully utilized.

So, let us say that W by T is greater than mu. If W by T is greater than mu, then the number of packets that needs to be buffered at the node will be equal to W minus B. So, B is W minus mu T or from this, W will become equal to P plus mu T; you can see from this relationship because if the source sending rate happens to be greater than the bottleneck links service rate, then W by T if it is greater than mu, then W minus mu T - these many packets need to be stored in the buffer and if W minus mu T exceeds the buffer size, there will be a packet loss, there will be a congestion conditions in the networks.

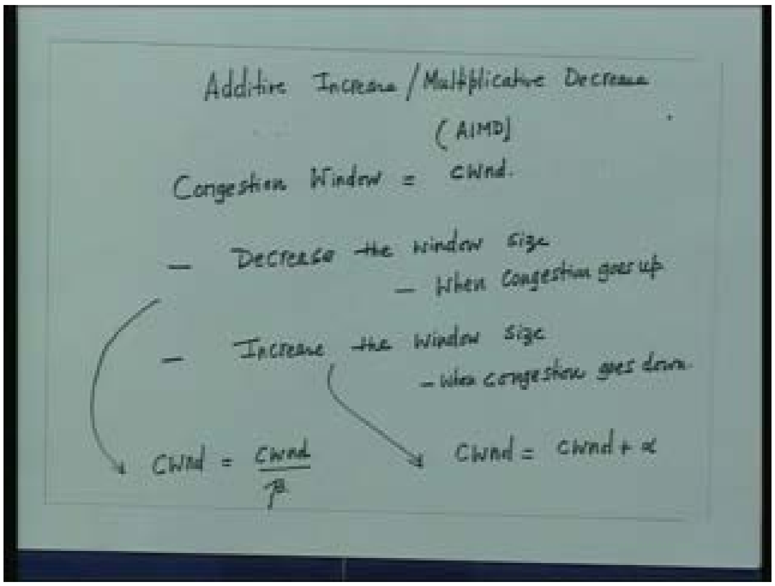
So, from this if you know what is the buffer size at a bottleneck link which is in this case B, we can determine how much is the maximum window size that we can have and depending upon how much is the bottleneck nodes queue occupancy, one can actually dynamically adjust the window size.

So, what we are saying essentially is that a maximum number of unacknowledged packets that can be there in the transit will be equal to the number of packets that can be stored in the buffer plus the number of packets that are there on the wire that is on the link. So, you add these two and that will give you what is the maximum number of unacknowledged packets because these many number of packets can then be there in the transit and therefore depending upon what are the queue occupancies at a different nodes, you can dynamically adjust the window size.

However, in practice of course you know that you do not know the exact queue occupancy or the buffer occupancy. All you will know is to adjust the window size is the feedback from the network which in some sense would be implicit and that implicit feedback will come to know only when a packet loss occurs or whenever you do not get an acknowledgment within a certain time of interval. And, therefore your timer will time out and then you will infer that since an acknowledgment has not arrived, it must have been due to the packet loss.

So, this explains the relationship that exists between the transmission window W and the bottleneck service rate.

(Refer Slide Time: 35:44)



So, now let us look what is a basic mechanisms of the TCP's congestion control that operates. And, the basic mechanism of the TCP's congestion control is based upon the additive increase and multiplicative decrease. It is also called as AIMD. Now, we define a certain variable which we call as the congestion window size. So, let us say that size of the congestion window is defined to be a variable which is CWnd.

Now, the principle of additive increase multiplicative decrease is that decrease the congestion, decrease the window size when the level of congestion goes up when congestion goes up and increase the window size and increase the window size when congestion goes down. Now, the question however is that since it is additive increase and multiplicative decrease; what we are saying is that whenever the congestion goes up, you decrease the window size in a multiplicative manner.

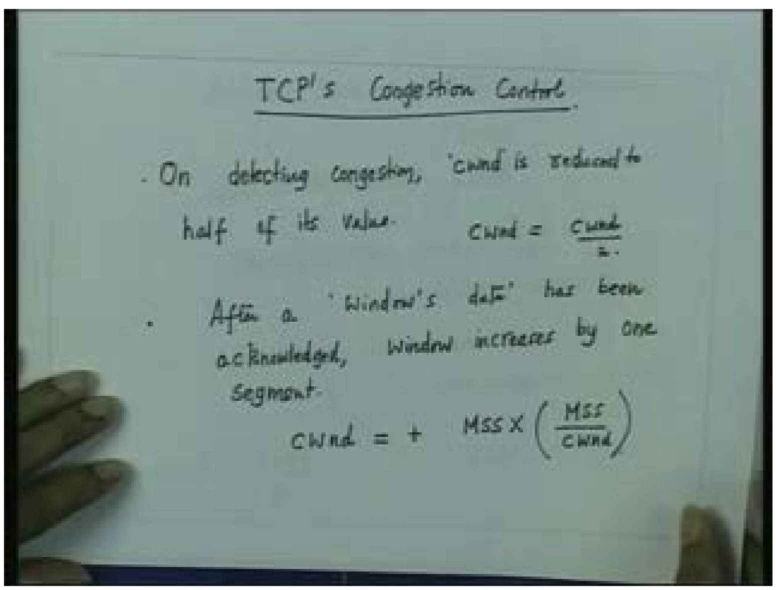
What is multiplicative manner? That you decrease the congestion window size if your current congestion window size is CWnd; you decrease the congestion window size to CWnd by some factor let us say beta. We will shortly see what are these betas and how actually the TCP's congestion control mechanism operates. But what I am trying to say that when I say that I am decreasing the window size in a multiplicative fashion, then I am reducing it by a certain factor which is a multiplicative factor of let us say beta. And, when I say that I increase the congestion window size, then I am saying that my new congestion window size becomes equal to the old congestion window size plus certain factor alpha.

So, my increase is in an additive fashion, my decrease is in a multiplicative fashion. So, my decrease is little faster, my increase is little slower. This congestion control mechanisms is called AIMD or additive increase multiplicative increase. So, the question obviously is that why are we not increasing the window size in a multiplicative fashion or why are we not decreasing the window size in a additive fashion?

The answer is that it has been found that the additive increase and the multiplicative decrease leads to a stable system in the networks and therefore that is what has been adapted that you increase the window size in a linear fashion, in an additive fashion and you decrease the and whenever you deduct the congestion, you decrease the window size by a certain factor in a multiplicative fashion. And as a result, this will lead to a stable system in the networks.

Now, the exact manner in which the TCP's congestion control mechanism operates, let me just explain that TCP's congestion control mechanisms.

(Refer Slide Time: 39:50)



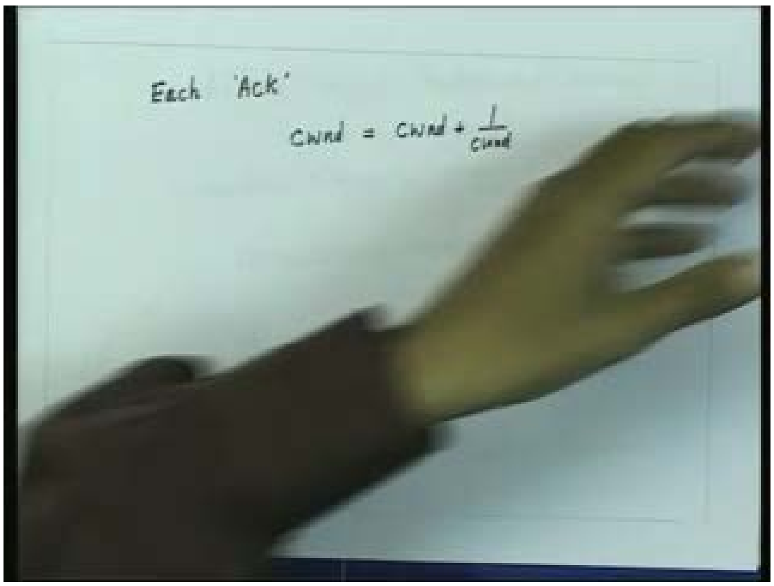
Now, the manner in which TCP's congestion control operates; now remember that a TCP's operates on a byte boundary that is a byte oriented protocol. So therefore, the window size in the TCP is measured not in terms of packets but in terms of bytes. So, later on for the purposes of analysis etcetera, we will assume that the window size is measured in terms of packets. But I just wanted to let you know that the TCP's window size is measured in terms of bytes. The sequence number and the acknowledgment numbers are with respect to the bytes and not with respect to the packets.

So, what happens in the TCP is that on deducting congestions, whenever the TCP deducts congestion and how exactly it deducts the congestions, I will just explain briefly later. But

suppose the TCP has deducted the congestions; then on deducting congestion the congestion window size that is the CWnd is actually reduced, the CWnd is reduced to half of its value. So, CWnd is actually equal to CWnd by half.
And, after a congestion windows worth, so a window's after a window's data, full window's data, all the packets has been acknowledged; the window will increase by one segment. That is in other words, CWnd increases by the maximum segment size multiplied by maximum segment size upon the congestion window.

Now, if you try to see this in terms of packets, when it is something like this that for every acknowledgment for every acknowledgment that comes, the window size increases by 1 by w and when all w acknowledgments have come, the window size will increase by 1.

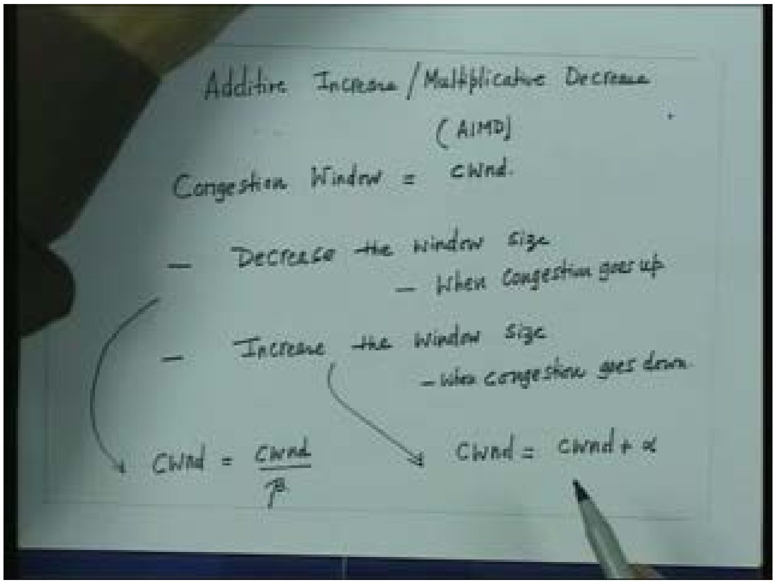
(Refer Slide Time: 43:13)



So, in a crude term we can say that every acknowledgment every acknowledgment, each Ack each Ack will adjust the window size, let us say CWnd is something like CWnd plus as measured in packets as measured in packets. So, if CWnd Acks have come, then the window size will by these by 1. But remember that the congestion window in TCP is measured in terms of bytes. So therefore, this is the correct equation where we are saying.

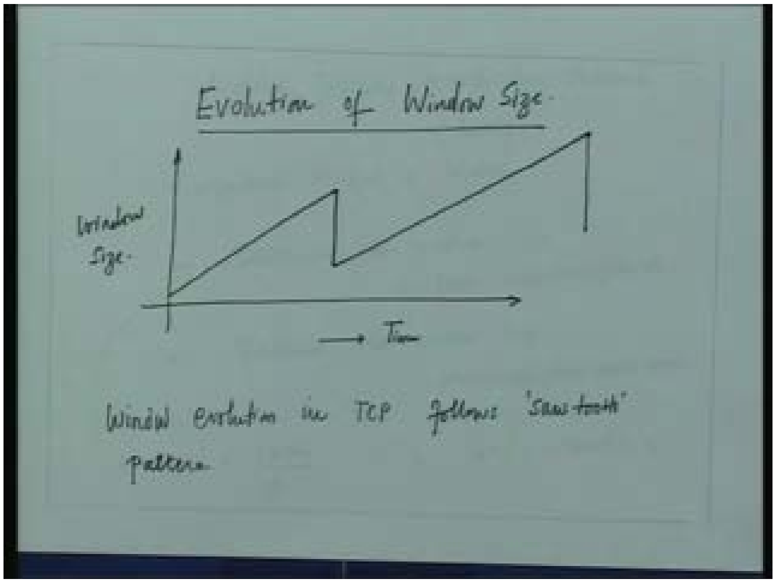
So, whenever you have received the acknowledgment for all the bytes that have been sent during the period of during CWnd, that means when acknowledgments for the entire CWnd worth of data has come, the window size will increase by 1 segment. And, whenever it deducts congestion, the window size will decrease to half. So, if you all see how therefore so therefore we say that congestion window is following an additive increase and multiplicative decrease.

(Refer Slide Time: 44:28)



You are decreasing in the window size by half; so beta is actually equal to 2 and you are increasing the window size by 1when the entire window has been acknowledged.

(Refer Slide Time: 44:42)



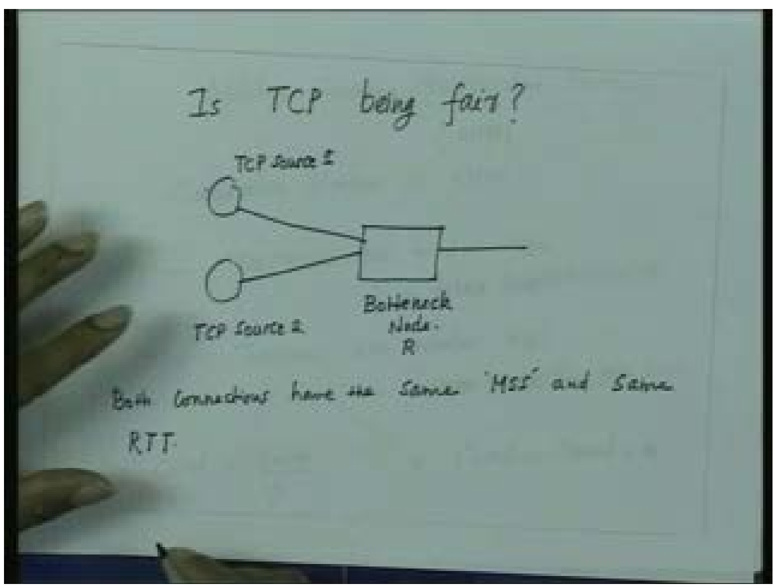
So, if you really see the evolution of the window size if you see evolution of window size in TCP, if you see evolution of window size in TCP, then your window size is increasing in a linear fashion, in an additive fashion and let us say, it deducts congestion here. Now, window size keeps on increasing by 1 after every time a window's worth of acknowledgment as received and whenever it deducts a congestion, the window size drops to half .

Similarly, again it increases the congestion window and this time it might reach upto here and whenever it deducts the congestions, it drops the window size to half and so on. So essentially, the evolution of the window size follows some kind of a saw tooth pattern. So, the window evolution in TCP, so this is like with respect to time and this with respect to window size. So, window evolution in TCP actually follows a saw tooth pattern. I mean you are increasing the window size, so again … the size that every time.

So, what is happening is that if W is the window size, then you will send W packets back to back and as you receive acknowledgment, you increase the window size by 1 by W for every acknowledgment. So, after you have received the acknowledgment for all the W packets, your window size will increase by one right because for every W, for every single packet; your window size is increasing by 1 by W. So, for W packets, your window size will increase by 1. So, this way your window size will increase by 1 at the end of receiving the acknowledgment for all the W worth of data.

If you deduct at a particular point that a congestion has occurred, the window size will decrease to half of its value. That is it will become W by 2. Now, the question that occurs that - is the TCP's congestion control mechanism operating it in a fair manner? Is the TCP's congestion control mechanism is fair? So, we ask this question that is TCP's congestion control is fair. Is TCP being a fair?

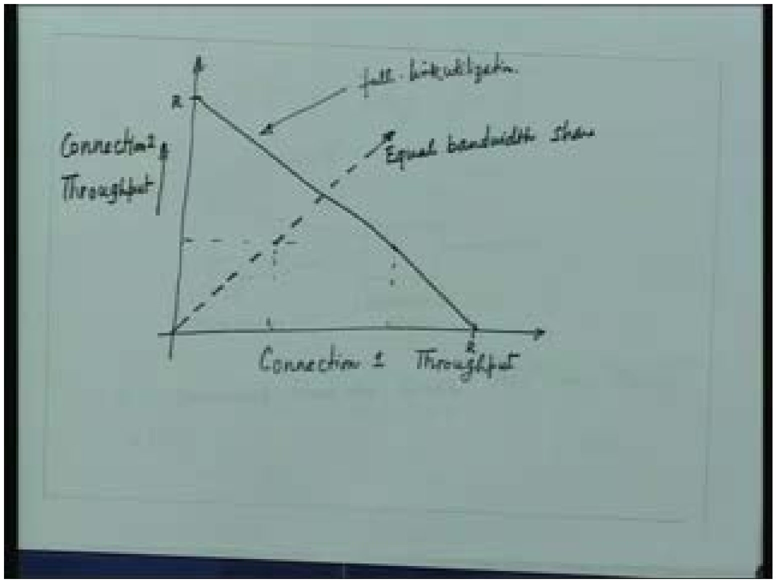
(Refer Slide Time: 47:43)



Now, let us assume that we are having a bottleneck node here through and there are 2 sources which are sharing this link; so source 1 sorry TCP source 1 and TCP source 2. So, these sources are sharing the … Now, assume that both these sources have the same round trip time and have the same maximum segment size, where MSS are the same. So, assume that both connections have same maximum segment size and same round trip time.

Now, we would try to see that whether the TCP's congestion control mechanism is operating in a fair manner or not.

(Refer Slide Time: 49:19)



So, let me just show you that suppose if we try to plot the two connections throughput, so this is connection 1 or TCP connection 1 throughput verses connection 2 throughput. Now, if you see if you plot this line R; since R is the output link and if I plot this straight line, if you take any point on this line let us say if I take this point, then clearly here the connection one is nearing higher throughput, connection 2 but sum of the 2 throughputs is R.

So therefore, this is the line if the point lies on this line that is when the connection 1 and connection 2 throughputs point lie on this line. Then we can say that a link is fully utilized. One connection may be getting a higher throughput and other connection may be getting a lower throughput. But the sum of the two rates or sum of the throughputs is always equal to R. So, this gives me a line which is getting full link utilization. So, this indicates full link utilization line.

Now, if I plot a line with 45 degree angle here, this line will give me a line with a equal bandwidth share. So, if I have some source here and so if I take a point here, then both the connections are having equal throughput. So, ideally I would like to achieve this point where both the connections are getting a throughput of R by 2, R by 2. That is how we would like to see.

Now, let us see whether the TCP's congestion control mechanism can achieve that or not. Now, suppose the 2 connections throughputs were in such a manner that their operating point was actually A. So, the connection 1 was getting certain throughput, connection 2 was getting certain throughput; the link was definitely underutilized because this point A lies below this and both the connections also not getting equal share.

Now, what happens is the TCP sources will start increasing their transmission rates the TCP's source will start increasing their transmission rates in parallel to the equal bandwidth share because both source, since both sources have assumed to be the same MSS and both sources have assumed to have the same round trip times; their rates will increase.

So, now their rates will increase in an equal fashion and till you… Now, when you become greater than this line, obviously you are congesting the length and let us say that the sources, both the sources determine that a loss of packet has occurred at this point B. Now, they dropped their rates to half and come to C. From C, again now it has become less than the link utilizations. So therefore, again the sources will start increasing their rates. Again, when you cross this line, you reach a point D, where again you deduct a congestions and so on.

Now, this from B, you have reduced the rate to half of your transmission rates. So, you can join the line from the origin to the B and reduce it to half of it. In the diagram of course, it is not drawn on 2 scale but what you can do is that you can join the line from the origin to the B and reduce it to half and in this way.

So, what we can see really is that a TCP, the 2 connections which are having the same maximum segment size in the same round trip time, they will try to achieve their optimal point. What is their optimal point? Their optimal point is this where both sources are getting equal bandwidth share and as well as the link remains fully utilized. So, this way we can have an effective allocation of the link bandwidths share.

Now, we can see that it is very easy to show, very easy to see that the TCP sources will achieve this point slowly. So, what they will do is by controlling this window size, so what they will do is that they will slowly try to know how much is the capacity in the network. So, they keep on increasing their transmission rates till the link is fully utilized and whenever the sources deduct a packet loss, they reduce their transmission rates and when the link starts becoming underutilized, it again start increasing the transmission rate and slowly because of these oscillations we approach the point where both sources are getting an equal approximately equal bandwidth share and at the same time, links remains fully utilized.

So, in some sense we can say that a TCP is fair. However, one important assumption that both sources have assumed to be having the same maximum segment size and the same round trip times. It can be easily shown that this argument will break down if the sources are having different round trip times or different segment sizes. If that is so, then we will require some kind of a packet scheduler at the nodes to isolate the flows with this kind of disparity having the different round trip times and therefore their different response behavior to the congestion condition in the network.