Broadband Networks

Prof A. Karandikar

Electrical Engineering Department

Indian Institute of Technology, Bombay

Lecture - 19

TCP Throughput

So, in the previous lectures we had derived the throughputs for the TCP in networks with high bandwidth delay products. So, we will continue that discussion and we know that the TCP has basically has 2 phases: one is the slow start phase and another one is the congestion avoidance phase. So, when we are trying to determine the throughput of a TCP in a network, then we need to determine the throughput in both - the slow start phase as well as that the throughput in the congestion avoidance phase and method that we had followed is that…

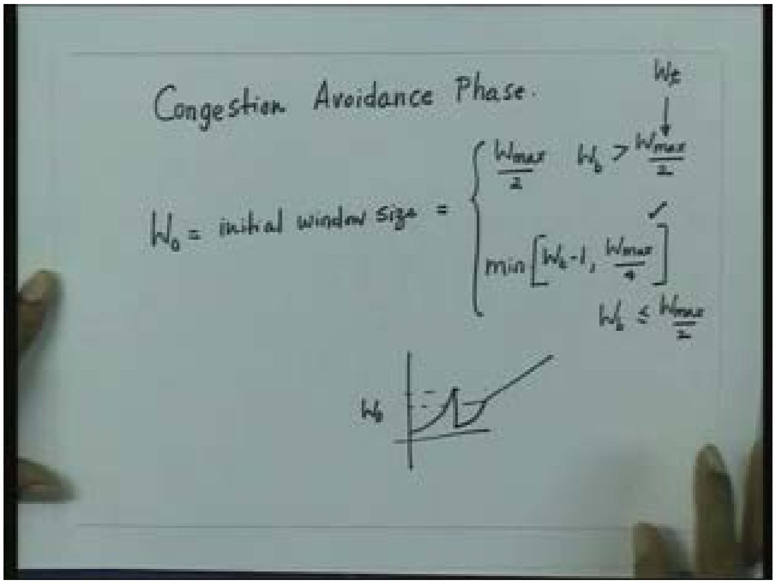
So, now there are 2 cases that can arise in the slow start phase. In one case, though buffer overflow occurs in the slow start phase and if no buffer overflow occurs in the slow start phase; then the window evolves upto the slow start threshold which is apriori set equal to half the maximum window size and from then onwards the congestion avoidance phase starts. So, if no buffer overflow occurs in the slow start phase, then the window evolves upto the slow start threshold variable and then, from then onwards the congestion avoidance phase starts and in the congestion avoidance phase if a packet loss is detected due to let us say triple duplicate acknowledgment, then the window drops to W by 2.

On the other hand, if the buffer overflow occurs in the slow start phase, so if a buffer overflow occurs in the slow start phase; then the window size drops down to W is equal to 1. That is it becomes 1 again and then again second slow starts phase starts immediately. But however this time, the slow start threshold variable is set equal to the half the window size at which the packet loss was detected in the slow start phase.

So, in both these cases - where the buffer overflow occurs in the slow start phase and where the buffer overflow does not occur in slow start phase; so there are two cases and we had analyzed both these cases and had determined the average cycle time and also the average number of packets that would be transmitted during the cycle.

Now, today what we will do is that we will try to analyze the congestion avoidance phase. Now, as we had said that since the congestion avoidance phase starts after the slow start phase, so since there were 2 cases of the slow start phase, there will also be therefore the 2 initial conditions for the congestion avoidance phase corresponding to the 2 cases of the slow start phase. So, let us see what those cases will be.

(Refer Slide Time: 4:19)

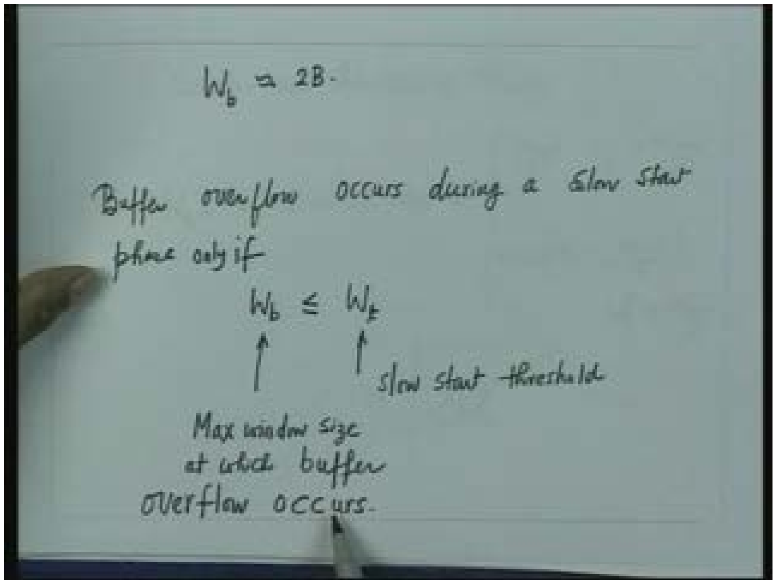


So, in the congestion avoidance phase, so we say that the congestion avoidance phase starts from the initial window size; so this is equal to the initial window size of the congestion avoidance phase and this initial window size that would be equal to W_{max} by 2 if W_b is greater than W_{max} by 2 and it would be equal to minimum of W_b minus 1, W_{max} by 4 if W_b is less than or equal to W_{max} by 2.

Now, this case corresponds to the case where or where no packet loss is detected in the slow start phase. That means this phase corresponds to the case where no buffer overflow occurs in the slow start phase and this case corresponds to the fact where the buffer over flow occurs in the slow start phase and since in this case the buffer overflow occurs in the slow start phase, the window evolves. So, it is something like this that the window evolves upto a certain point here and then the window drops to 1 and then again it goes upto this point and from then onwards, the congestion avoidance phase starts.

So, for this second case, the congestion avoidance phase W_0 will become equal to minimum of W_b minus 1 or W_{max} by 4. Now, the definition of W_b, we have already seen. So, W_b is the window size, the maximum window size at which the buffer overflow at which the buffer overflow will occur. So, let me just recap the definition of the W_b.

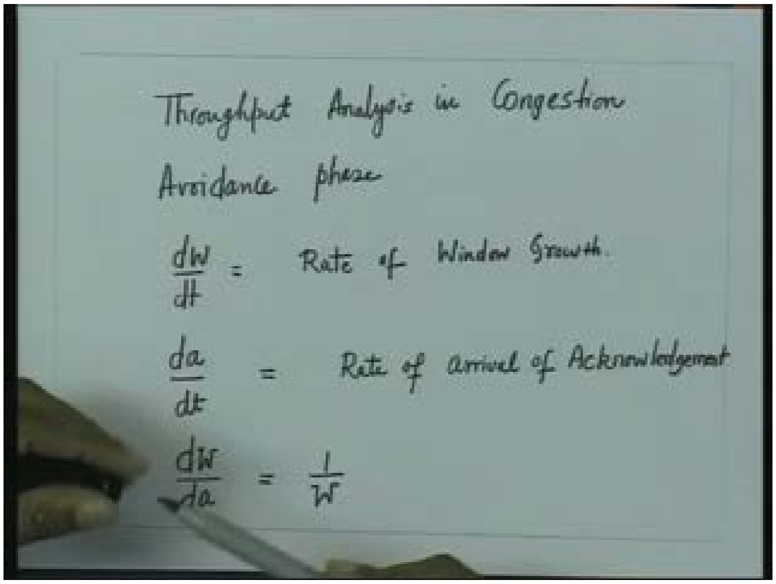
(Refer Slide Time: 6:35)



So, we had seen W_b is the maximum window size at which the buffer overflow occurs and we had seen that approximate value of W_b will be equal to twice the bottleneck buffer size. So, that is how we had defined W_b. So, if W_b is greater than W_{max} by 2 which was the initial slow start threshold, this was actually the initial slow start threshold that is W_t; if that is so, then no buffer overflow occurs in the slow start phase and the initial window size in the congestion avoidance phase is W_{max} by 2.

However, if W_b is less than or equal to W_{max} by 2 that means the maximum window size at which the buffer overflow can occur happens to be less than the slow start threshold variable and therefore the congestion avoidance phase will start from here. So now, we will try to analyze the throughput in the congestion avoidance phase.

(Refer Slide Time: 7:30)



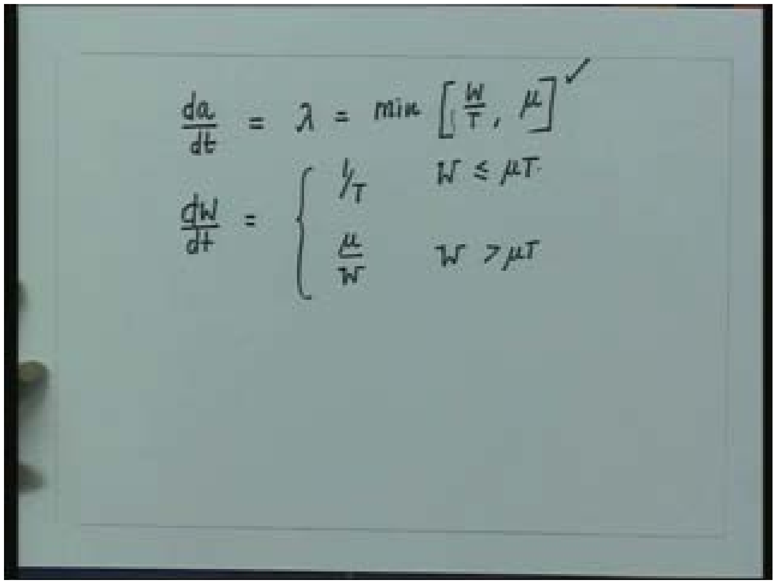
So, what we will do is that we will do the throughput analysis in congestion avoidance phase. Now, what we do is that even though we know that the window evolution takes place in a step like fashion, in the sense that whenever an acknowledgment comes, the window size increases by 1 by W in the congestion avoidance phase.

So, when the acknowledgments for all the W packets have arrived, then the window size actually increases by 1 in the congestion avoidance phase. So, now the acknowledgments for all the packets sent during window size is expected to arrive within the round trip time. So, what happens? Within a round trip time, the window size actually increases by 1. So, the window size actually follows a step like increase.

However, for the simplicity of the analysis we assume that the window is increasing in a linear fashion and therefore as we had discussed in the previous lecture also, the window evolution is assumed to be having a saw tooth pattern in the congestion avoidance phase. So, we are assuming it to be a continuous linear increase and whenever the packet loss is detected, the window drops to W by 2 and so on.

So now, let us define the quantities like dW by dt which is the rate of increase of the window size, dW by dt is the rate of window growth and let us define da by dt which is the rate of arrival of acknowledgment rate of arrival of acknowledgment. Now, we know that dW by da that is the rate of window growth with respect to the acknowledgment arrival is actually given by 1 by W. Now, this is because as we have seen that the window size increases for every acknowledgment, the window size increases by 1 by W. So, therefore dW by da is actually 1 by W.

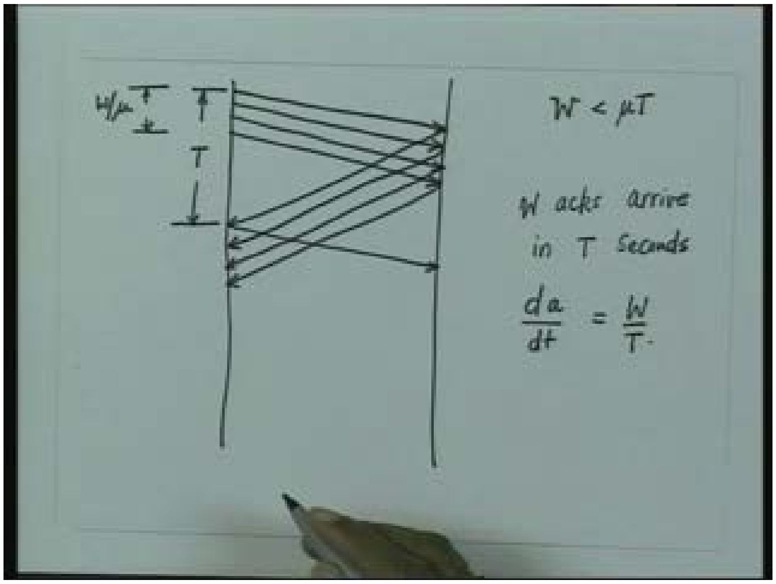
(Refer Slide Time: 10:33)



Now, da by dt that is the rate of arrival of acknowledgment and let us denoted by lambda; this is actually minimum of W by T into mu where T is the round trip time and mu happens to be the bottleneck service rate. So, that is da by dt.
So, we know that dW by da is 1 by W, da by dt is minimum of this; so, in that case then, dW by dt that is the rate of window growth that will be equal to 1 by T if W is less than or equal to mu T and it will be equal to mu by W if W is greater than mu T.

Now, that depends upon if W by T is actually happens to be greater than mu then, da by dt will be mu and then dW by dt will become mu by W. However, if this happens to be minimum, then da by dt will be W by T and dW by da is 1 by W. Then, we will get dW by dt is 1 by T. Now, let us try to understand how we have arrived at this result. Just to understand this result, I will just explain this with a diagram.

(Refer Slide Time: 12:07)



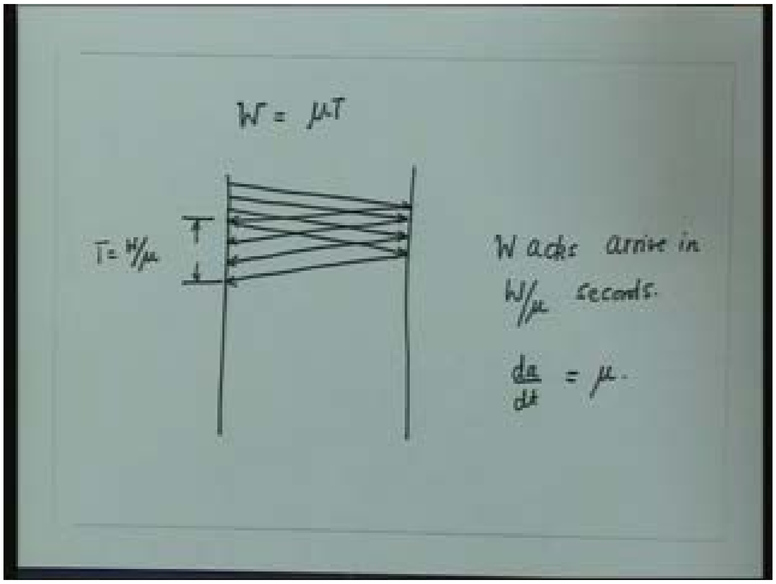
Now, let us say this is a sender and this is the sort of receiver and so now when we send, let us say the window size is 4; so when we send 4 packets – 1, 2, 3, 4, let us consider the case where W is less than mu T. So, the window size is actually happens to be less than the delay bandwidth product.

Now, what happens? The acknowledgment actually starts arriving after the round trip time T. So, if you start at the time here; then after the T time interval, this acknowledgment T will actually arrive and after 1 by mu times; I will receive the acknowledgment for this third packet and I will receive the acknowledgment of the fourth packet.

So now, this is the total window size W, remember and the distance between the separations between the two packets is actually 1 by mu that is the service time. So therefore, if you consider this time, this time happens to be equal to W by mu. Now, you can see that in this case when W is and so on and from this you will start transmitting, then one packet and so on. The window size increases by 1.

In this case, you can see that approximately W acks, they will arrive in T seconds approximately. So therefore, da by dt happens to be following of W by T. So, that is how we will get this result that da by dt will be equal to… If W by T happens to be less than mu, then it will be equal to W by T.

(Refer Slide Time: 14:36)



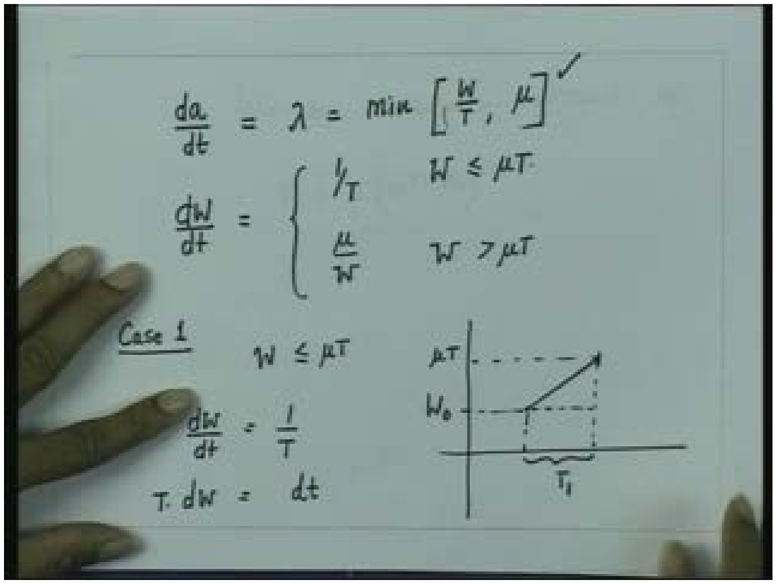
Now, let us consider a case where let us say W is equal to mu T. In this case itself, you consider the W is equal to mu T. Now when that happens, our scenario changes something like this 1, 2 third packet and the fourth packet.

Now, by the time you have sent, you started sending the fourth packet; your acknowledgment for the first packet should have arrived and then after mu second, this acknowledgment should have arrived and after another 1 by mu second, this acknowledgment should have arrived and so on. So, as a result you can actually see this window, this happens to be equal to T is equal to W by mu second. So, here what happens is that W acknowledgments, they arrive in W by mu seconds and therefore da by dt happens to be mu and even if it is greater, even if W is greater than mu T; then even da by dt will remain mu.

So, that is how we have arrived at this relationships that da by dt is actually happens to be minimum of either W by T or mu. So, what we are actually trying to demonstrate is that how the growth of window evolution and how the growth of arrival of acknowledgment takes place in the congestion avoidance phase. So, now we have got the basic deferential equation of the basic deferential equation governing the dynamics of the window evolution in the congestion avoidance phase.

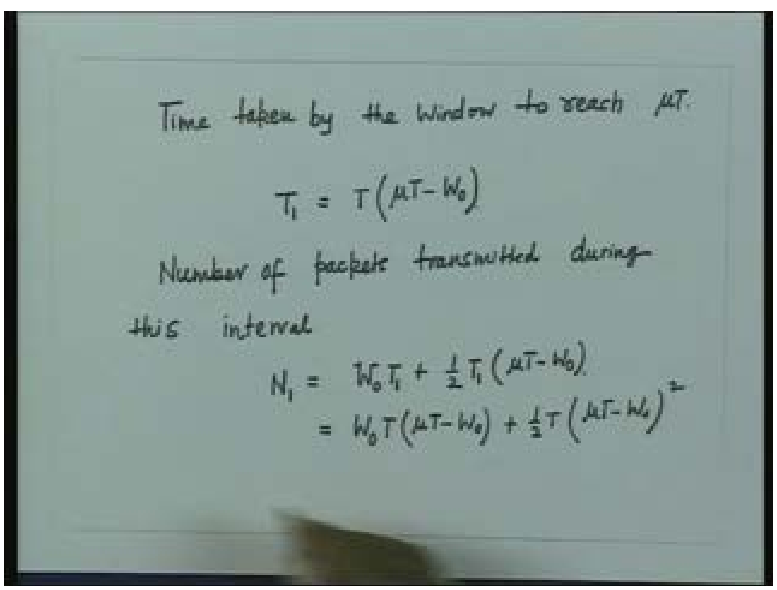
So now, let us look at this basic deferential equation which is W by T is equal to 1 by T and mu by W. Now, consider the first case.

(Refer Slide Time: 16:56)



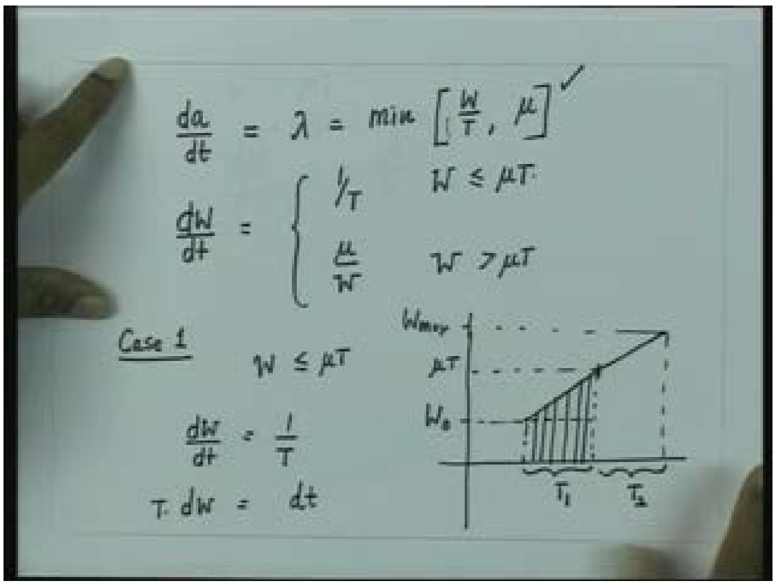
So here, we will consider let us say the case one that is W is less than or less than or equal to mu T. So, what we are assuming is that the window size is actually less than or equal to mu T. In that case, dW by dt is given by 1 by T. So, if I plot, so you start the window from W_0 ; before this you are having the slow start phase, you increase the window and increase the window and W is less than or equal to mu T ; now, let us say that the window size becomes equal to mu T. Till this period, the window evolution the window evolution is followed by dW by dt equal to 1 by T and let us say the time taken to reach this is the duration of that interval let us say T_1. So, if this is so, dW by dt is equal to 1 by T.

(Refer Slide Time: 18:05)



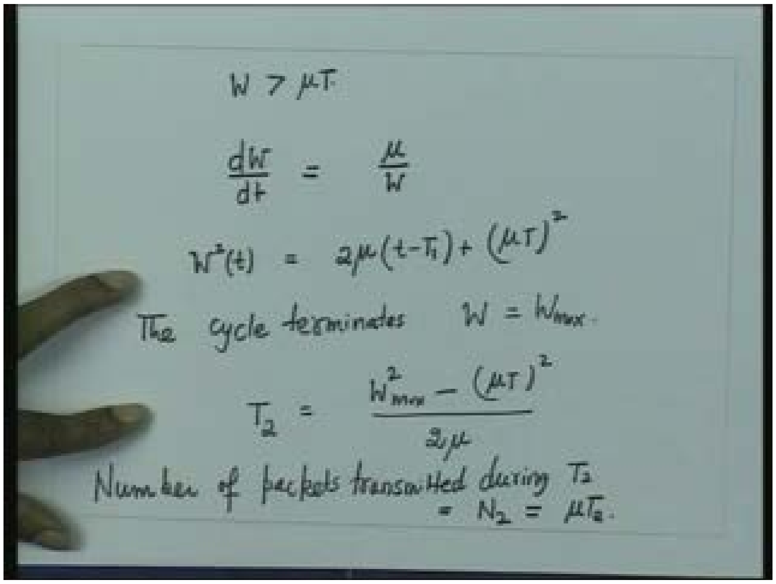
So, we need to determine what is the time taken, what is the time taken by the window to reach the size of mu T and that is given by T_1 is equal to T times mu T minus W_0. This is obtained by simply solving this deferential equations which means the T into W is actually equal to dt and we integrate this for the dW; the limit happens to be from W_0 to mu and for the dt this is from 0 to T_1 if you assume the time to be here equal to 0. So as a result, we have shown that a time taken by the window to reach mu T is equal to this and and the number of packets and number of packets transmitted during this interval, now number of packets transmitted during this interval as we have seen is the area under this curve.

(Refer Slide Time: 19:28)



So, number of packets transmitted during this interval is given by the area under this curve that will tell you how many number of packets have been transmitted during the congestion avoidance phase and that is given by N_1 equal to W_0 T_1 plus 1 by 2 T_1 into mu T minus W_0 and we can substitute here the value for T_1 and we will get as W_0 T mu T minus W_0 plus 1 by T mu T minus W_0 square. So, this comes immediately after this. Now, when the window size reaches this mu T; then after this, this equation is valid. That is dW by dt is equal to mu by W.

(Refer Slide Time: 20:40)



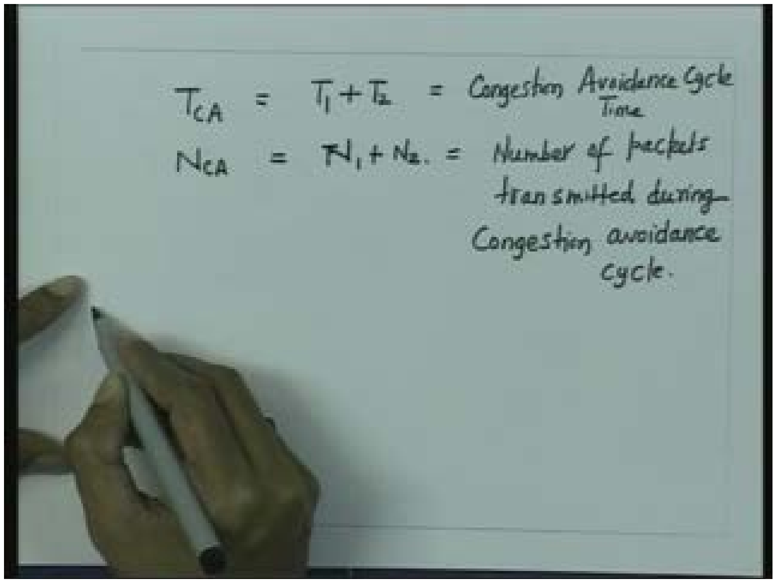
So, the second phase, the second phase starts when the window size W is greater than mu T. Then the governing equation is dW by dt to be equal to mu by W and let us assume that this window reaches W $_{max}$ and to reach this, it requires another time T_2. Now, during this interval which equation is governing? This equation - dW by dt is equal to mu by W. So, we need to solve this equation with the initial conditions of the initial conditions of mu T and the final is W_{max}.

So therefore, during this conditions, we get, if you differentiate this equation W square T will be equal to 2 into mu t minus T_1 plus mu T square where t denotes the time between time during this interval. Now, the cycle terminates when the cycle terminates when window size W becomes equal to W $_{max}$ and therefore the T_2 interval is given by W square $_{max}$ minus mu T square upon 2 mu. Because, then suppose this is the time interval which is time T, let us say T prime; so when T becomes equal to T prime and here we have assumed the time to be equal to sort of 0, then T prime minus T_1 that becomes equal to T $_2$. So, that is how we determine the time.

Now, the number of packets which are transmitted during this interval; since the link is fully utilized, since W is since during this case W is greater than mu T, the link is fully utilized and all the packets are being served with the bottleneck service rate mu and the cycle last for a total time period of T_2, the number of packet which are transmitted will be simply given by mu into T. So, the number of packets transmitted during T_2 the number of packets transmitted during T_2 which is N $_2$ which is equal to N $_2$ is given by mu T $_2$.

So now that so now what we have essentially done is that we have determined the cycle time of the slow start phase and the cycle time of the congestion avoidance phase and have also tried to determine what is the number of packets which are transmitted during the slow start phase and during the congestion avoidance phase in one cycle. And, so if you divide the number packets transmitted during one cycle and divide by the total cycle time, we will get the throughput. So, let us see what is the total duration of the congestion avoidance phase.

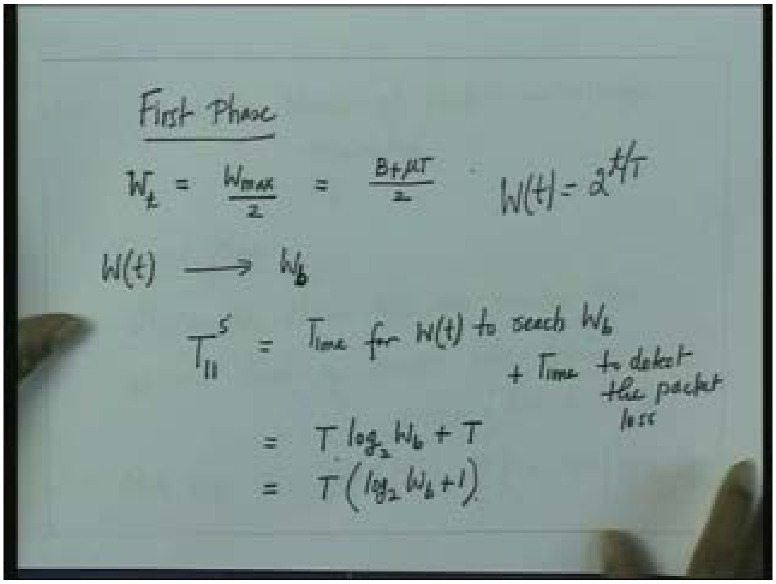
(Refer Slide Time: 24:33)



The duration of the congestion avoidance phase as we have seen, T of congestion avoidance phase is actually T_1 plus T_2 - this is the duration of the congestion avoidance phase and N_{CA} that is the number of packets transmitted is N_1 plus N_1. What about so, this is the congestion avoidance duration, congestion avoidance cycle time and this is number of packets transmitted during congestion avoidance cycle.

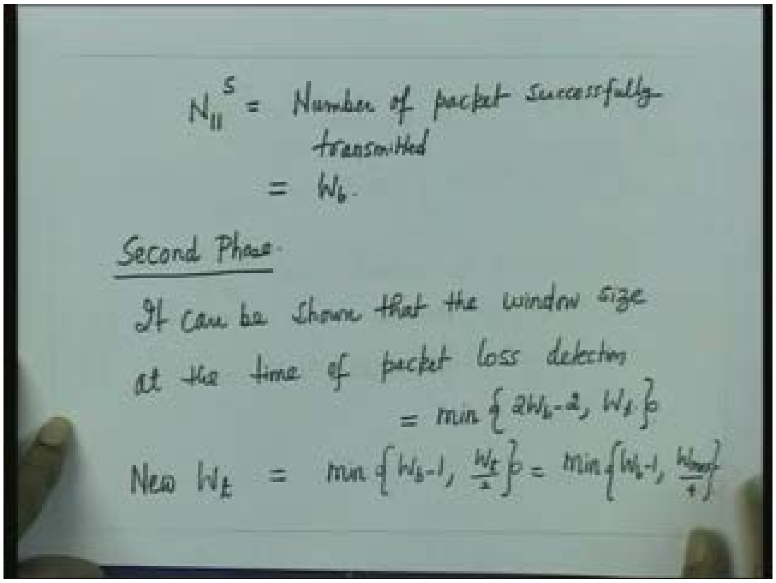
As far as the slow start slow start threshold is concerned; we know similarly that in the case of slow start, we had determine the number of packet transmitted during the slow start threshold and which was this.

(Refer Slide Time: 25:54)



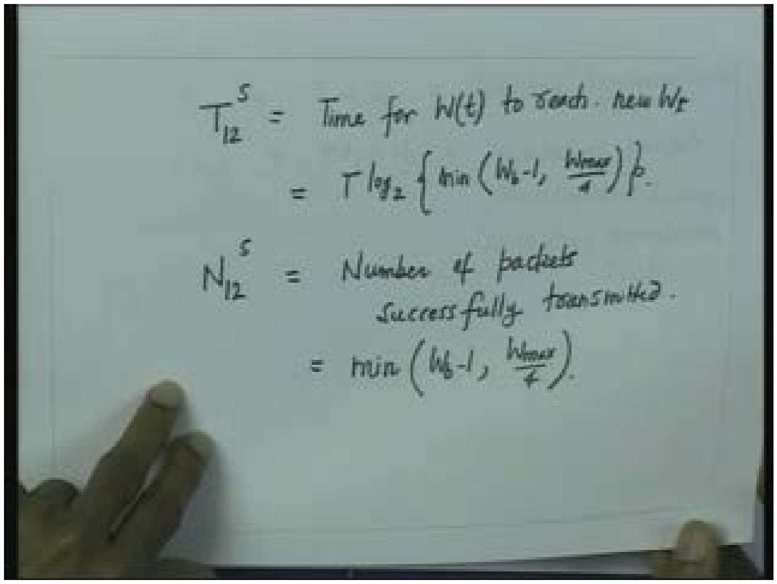
So, in the case where the buffer overflow occurs in the slow start phase; there were 2 phases. So, T_{11}^{S} was the time taken for the Wt to reach W_b and this was one cycle.

(Refer Slide Time: 26:11)



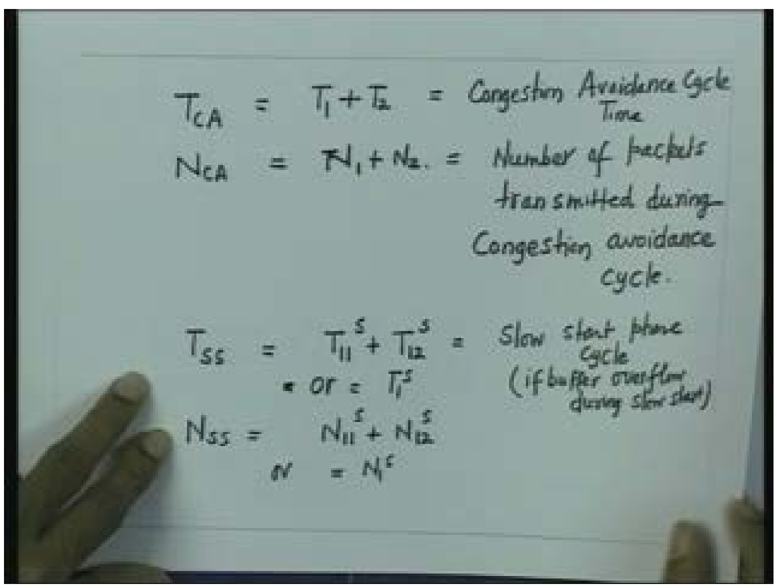
And then, we determine apart from T_{11}^{S}, we determine the N_{11} which was the number of packets successfully transmitted.

(Refer Slide Time: 26:18)



And then, we determine T_{12}, the second slow start phase and the number of packets transmitted during this phase.

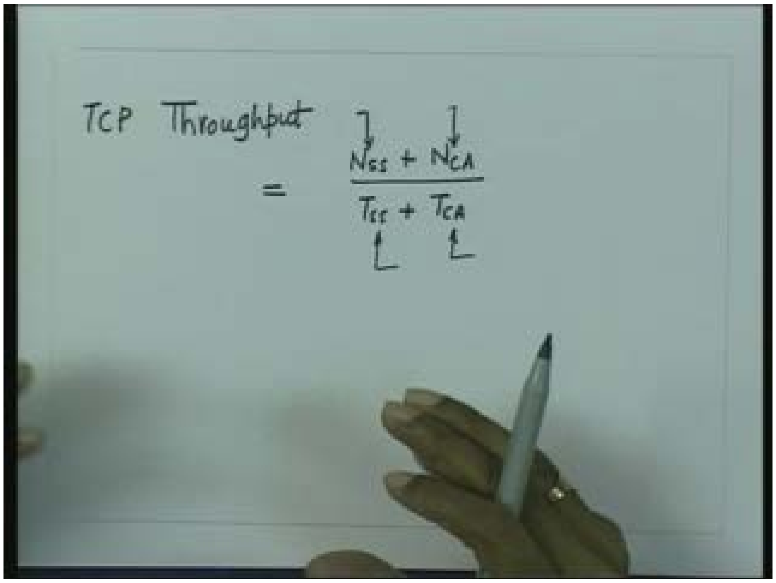
(Refer Slide Time: 26:26)



So therefore, the slow start the slow start phase, so I say a slow start phase, it will be given by T_{11}^{S} in the first T_{12}^{S} and this is equal to the slow start phase cycle. In the case when the buffer overflow occurs, if buffer overflow occurs during slow start phase and N_{SS} is given by N_{11}^{S} plus N_{12}^{S} that we have already determined and this is the number of packets which are transmitted during the slow start phase.

So therefore, the throughput and of course, if the buffer overflow does not occur during the slow start phase; then also we have determined the cycle time in the slow start phase and the number of packets which are successfully transmitted during the slow start phase. So therefore, the throughput otherwise, this will be equal to just or it will be equal to T_1^S or it will be equal to N_1^S that we have already determined.

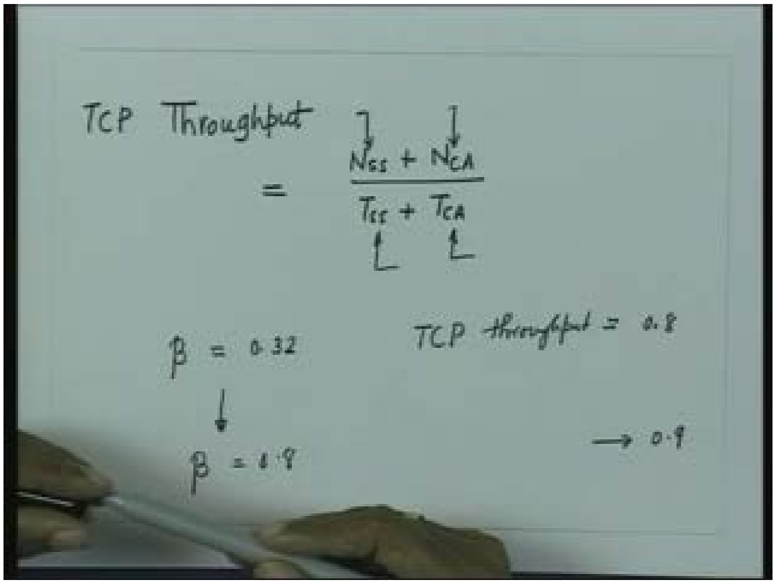
(Refer Slide Time: 27:54)



And therefore in that case, the throughput of a TCP throughput expression is given by N_{SS} plus N_{CA} divided by T_{SS} plus T_{CA}. That is the number of packets transmitted during the slow start phase, the number of packets transmitted during the congestion avoidance phase divided by the slow start cycle time plus the congestion avoidance cycle time; we will get the average throughput.

Now, if you numerically if you numerically compute these throughput for various values of beta, we can determine the numerical values of the throughput that would exist for determining the TCP throughput in a network with high bandwidth delay product. So, this gives an insight into determining an expression for the TCP throughput.

(Refer Slide Time: 29:04)



In fact, we can see that if we try to compute this expression for various values of beta, we can see that when beta is approximately equal to let us say 0.32, the TCP throughput is approximately 0.8 and after that if beta is even 0.8, even if you increase; then the TCP throughput goes to 0.9 only.

So, you can see that a threshold effect takes place at beta approximately equal to 0.33 where we had already shown that in a high bandwidth delay network products, a threshold effect exists at the window size for the normalized buffer size beta to be equal to 1 by 3 that is 0.33. So, this results therefore gets confirmed if we try to determine expression for the TCP throughput.

Now, in this case we have analyzed where we have considered simplified view, where we have considered source destination pair single TCP connection and a single bottleneck link and we have considered a case of a high bandwidth delay networks. So, that means the bottleneck service rate multiplied by the delay happens to be high and under that situations, we had tried to determine what will be the values of the throughput in the slow start phase and in the congestion avoidance phase.

Now, in the next analysis that we would like to show is an expression for determining the TCP throughput when we consider the random losses. So, that is another way of determining the TCP throughput when the losses occurs randomly with certain packet loss probability which is p. Note that a simplified analysis of the TCP throughput when the packet loss occurs with the probability p, we have already discussed and we have shown that the TCP throughput is inversely proportional to the round trip time and it is also inversely proportional to the square root of the packet loss probability

So, this is an important result, an expression for the TCP throughput when the packet loss occurs randomly. So, the TCP throughput is inversely proportional to the square root of the packet loss probability and that analysis we do we did with a very very simplified assumptions. We will try to refine this analysis in today's lectures and see what is the exact expression for the TCP throughput and we would also try to show that that expression is actually degenerates to the expression for the TCP throughput analysis that we had done in our previous lectures where the TCP throughput was shown to be inversely proportional to the square root of the packet loss probability.

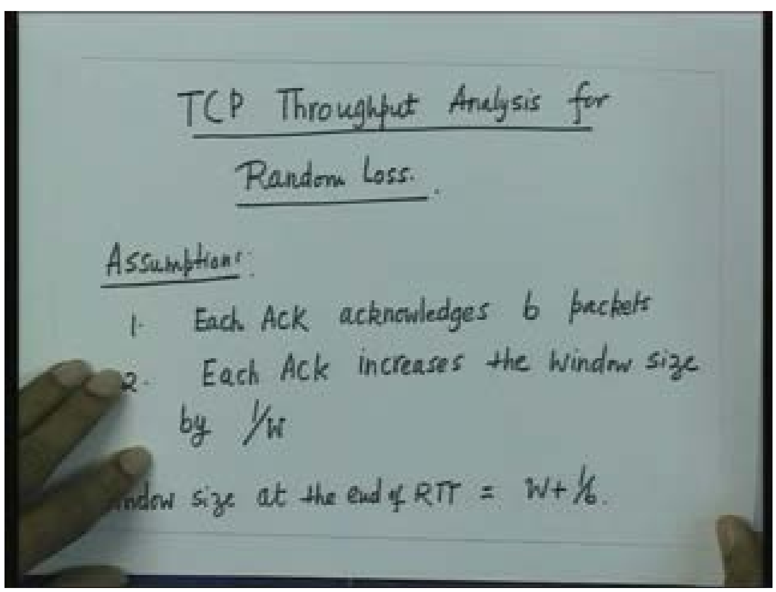
So, let us try to do that analysis and that analysis we will do only for the congestion avoidance phase. Now, remember that this particular analysis that we had done for the congestion avoidance phase is we have not assumed the random loss, we have assumed that a pipe exist and the pipe has bottleneck node which has a buffer capacity of b and the bottleneck service rate happens to be mu and the round trip time is T; so therefore, the maximum number of unacknowledged packets that can be injected into the network will be equal to b plus mu T.

Why? because, the bottleneck buffer has the capacity of storing b packets and the mu T packets can be in the transit. Therefore, the total number of unacknowledged packets that can be there in the network will be equal to b plus mu T and therefore that will be equal to the maximum window size and if you try to inject packets more than this window size, then a buffer overflow will occur. So, this was our assumption in the analysis.

Now, what we will try to do is that we will not assume about the buffer capacities in the bottleneck node but we will try to assume that a packet loss occurs at some bottleneck node with a probability of p and then determine an expression for the TCP throughput as a function of the packet loss probability p.

So, let me just draw a sketch of that analysis. We will make certain assumptions and then try to see how do we carry out the analysis of the TCP throughput.

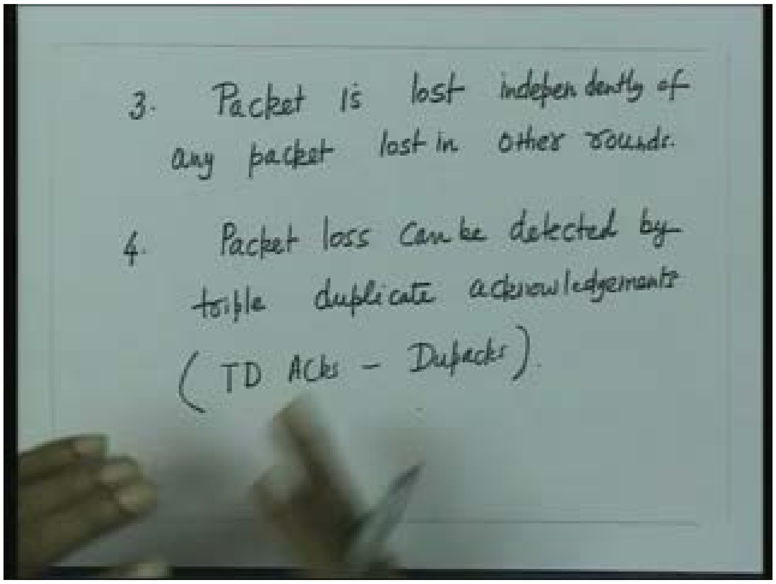
(Refer Slide Time: 33:43)



So, this we do as a TCP throughput analysis for random loss. Now, this has certain assumptions and the assumptions are - we assume first of all that each acknowledgment acknowledges b packets, actually till now we have done the analysis where b was assumed to be equal to 1, typical number of b could equal to 2 and these are cumulative acknowledgments. Then, and of course, each Ack as we had assumed, increases the window size by 1 by W.

Now, total number of acknowledgments that they will arrive… Now, remember that each acknowledgment is acknowledging b packets and in a window size of W, we are sending total W packets. So therefore, the total number of acknowledgments that will arrive will be W by b and therefore at the end of a round trip time, the window size will increase by W by b into 1 by W. So therefore, it will increase total by 1 by b. So therefore, the window size at the end of 1 round trip time will be W plus 1 by b. This is an important assumptions that each acknowledgment each acknowledgment is acknowledging b packets.

(Refer Slide Time: 36:15)

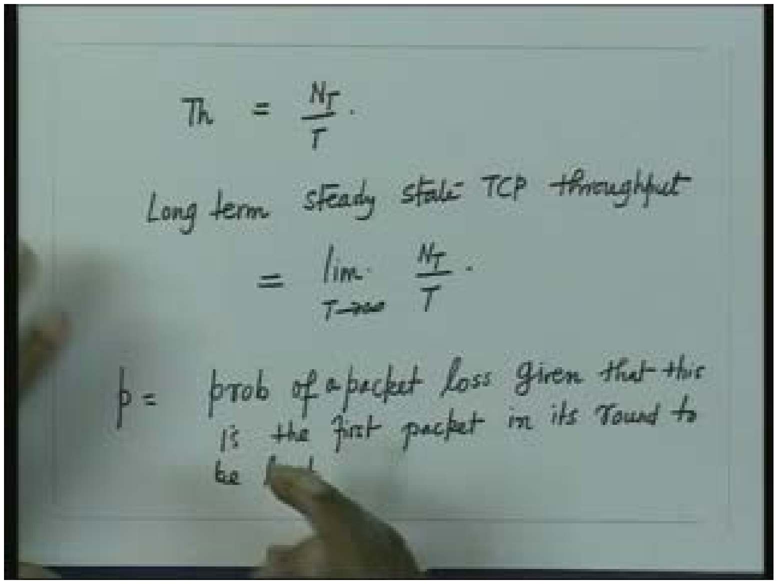


The second thing that we assume is that packet is lost in around independently of any packet lost in other round. Round means; so you have a window size of W, we are sending these W packets back to back and when these W packets reach the destinations, the acknowledgment arrive and this takes 1 round trip time. This we call it to be a round. So, in 1 round what will happen is that W by b acknowledgments will arrive and the window size will increase by 1 by b. So, this is what is called as 1 round.

We are assuming that packet is lost independently of any packet lost in other round and also if a packet is lost, then all other packets following that in that round are also assumed to be lost. So, packet loss in different rounds had happened independently but if in a particular round, if a packet loss occurs, then all other packets following that packet will be considered to be lost and in this analysis, we are assuming that packet loss can be detected by triple duplicate acknowledgment; so, what is commonly called as TD Ack or Dupacks.

We also make an assumptions that receiver's window size is not limited by the receiver's advertised flow control window; so, just to make our matters simple in terms of the window evolutions.

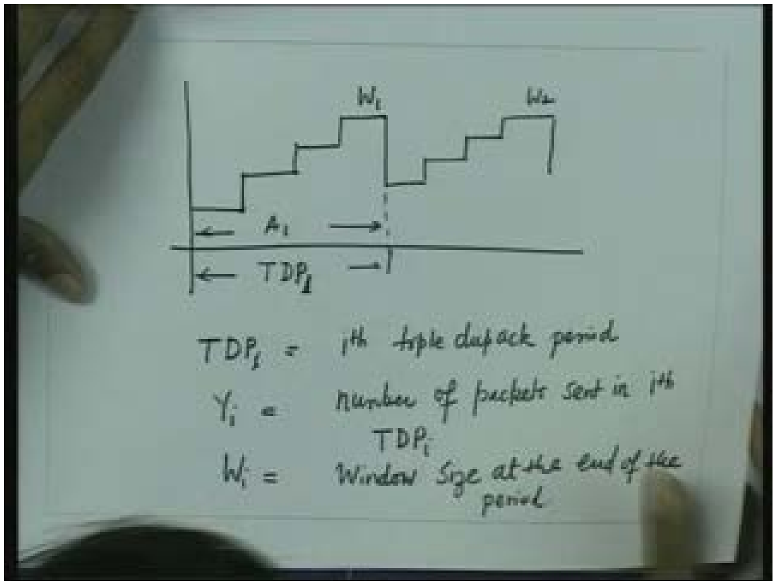
(Refer Slide Time: 39: 46)



Now, we define again in T to be the number of packets transmitted in the interval of 0 to t. So, this we call is N_T transmitted during the interval of 0 to T; then the throughput, then the TCP throughput per unit time that is the number of packets transmitted per unit time is throughput is given by N_T upon T and we call as the long term steady state TCP throughput will be assumed to be limit T tends to infinitive N_T by T.

So now, with this, the long we have we have determined the long terms steady state throughput and let p denote the probability of packet loss; given that probability of a packet loss given that this is the first packet in the round to be lost in its round to be lost the that means it is the probability of a packet loss given that no other packet is lost in this round. So, that is what the assumption is.

(Refer Slide Time: 41:05)

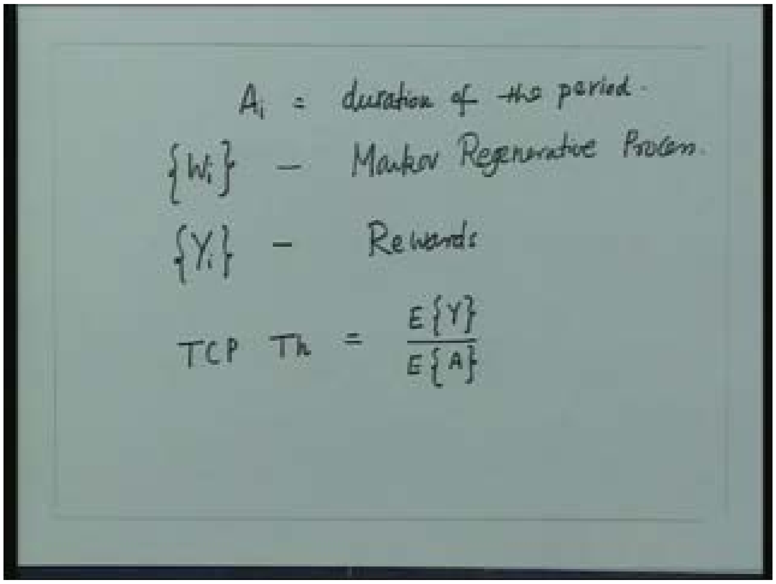


Now, so the window evolution takes place something like this. So, what we are assuming here is that the window; so this is like increase in window by 1 and so on. So therefore, this is equal to 2 rounds if b is assumed to be equal to 2 or if it is b, then it is b round. So, in b rounds the window will increase by 1. So, we are assuming a stair case function.

And now, at this point we assume that a packet loss is detected due to triple duplicate acknowledgment. So, this we call it to be a triple duplicate acknowledgment in the i'th period. So, this is the window size to be equal to W_1 and this is W_2 and so on and this is TDP_1 TDP_2 and so on.

So, we say that define TDP_i to be the i'th triple Dupack period. Let us say Y_i is equal to number of packets sent in i'th triple Dupack period that means in i'th triple duplicate acknowledgment and W_i happens to be the window size at the end of the period.

(Refer Slide Time: 43:13)



And A $_i$, so this A $_1$ which is happens to be the duration of the i'th period; so we say that A $_i$ is the duration of the i'th triple duplicate acknowledgment. Now, it is easy to show that the W $_i$ which is the window size at the end of i'th period is a Markov process, it is a Markov chain, it forms a Markov chain and moreover it is considered to be a Markov regenerative process.

Now, the regenerative process means the cycle repeats. So, really speaking if you consider the sequence which is W $_1$ W $_2$ W $_3$ and so on, it forms a Markov chain because the window size at the end of the period really depends upon what was the window size at the end of the previous period because that is the initial window size and whatever the number of packets that arrived during this period.

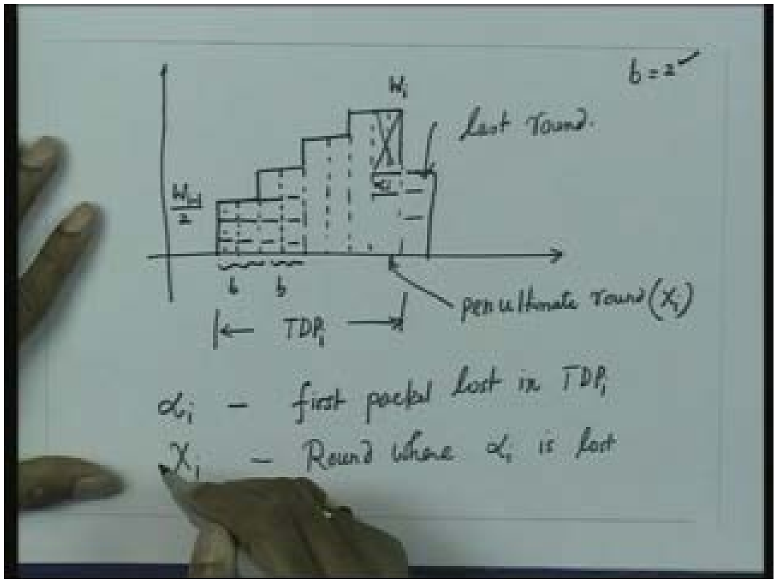
And since, the packet loss is assumed to be independent of the packet losses that occurred in the previous cycle; so therefore the W $_i$ happens to be a Markov process. It is also a Markov regenerative process and it earns a reward and these rewards are denoted by Y $_i$ which is actually the number of packets sent in this period. So, they constitute to be the reward.

So, if you apply the reward renewal theory, then it is easy to see that the throughput that is the number of packets which are transmitted per unit time is given by the TCP throughput - Th will be given by expected value of Y $_i$ divide by the expected value of A which happens to be the duration of this the regenerative cycle.

So typically, if you want to determine how many number of packets transmitted per unit time during this cycle; so we will say that number of packets transmitted is Y $_i$ divide by the cycle time which is A $_i$. But this is a Markov regenerative process, so we can say that long term steady state throughput will be given by expected value of Y that is expected value of number of packets transmitted during this period divide by the expected value of the period itself. So, this way we can determine the TCP throughput.

So, our objective therefore is to determine an expression for the expected value of Y and expected value A. So, this is what our objective is and this is what we will try to determine.

(Refer Slide Time: 45:51)



So now, let us concentrate our attention on the specific window evolutions that we had seen. So, let us say that this i'th TDP period, so we start so for the in this diagram assume that b equal to 2. So, it happens something like this; so we start from W_i minus 1 by 2. So, the window size increases like this, the window size is increasing; so as you can see here that the window size here was 3 let us say W_i minus 1 by 2. So, we are sending packets 1, 2, 3 and at the end of.

So, this is 1 round. So, at the end of this round, an acknowledgment will come and the window size will increase by 1 by b and in this case, b is 2. We have assumed that it will increase by 0.5 and after 2 rounds the window size will actually increase by 1. So, here we can say that this is b and this is b and so on and here, the window size increases. I have shown it to be by 1.

So, this is 1 triple Dupack period and let us say W_i is the maximum window size at which it reaches and packet loss is detected. So now, let us say that this is the last round we assume. So, this is the pen ultimate round, in the sense that a packet loss occurs in this round. So, let us say that alpha$_i$ happens to be the first packet which gets lost, first packet lost in this triple Dupack period and let us say that X_i is the round where this packet loss occurs, round where alpha$_i$ is lost. And, this is the pen ultimate round X_i and let us say that this packet alpha$_i$ is lost here and after this all the consecutive packets are lost. So, all these packets are lost.

Now, when the triple duplicate acknowledgment takes place, by that time in the last round, some more packets also have been transmitted. So, these many packets have already been transmitted and the receiver will receive the indication that a packet loss is occurred only when these many packets have already been sent. So, this is like the last round.

Now, what we will try to do is that we will try to determine what is the number of packet's expression for Y_i that is the number of packets transmitted. We will determine an expression for A_i that is the cycle duration and then we will determine their expectation values and from that we can easily determine the TCP throughput. So, we will discuss this how to determine the value of Y_i and how to determine the value of A_i.

(Refer Slide Time: 50:02)