Computer Networks and Internet Protocal
Prof Soumya Kanti Ghosh
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

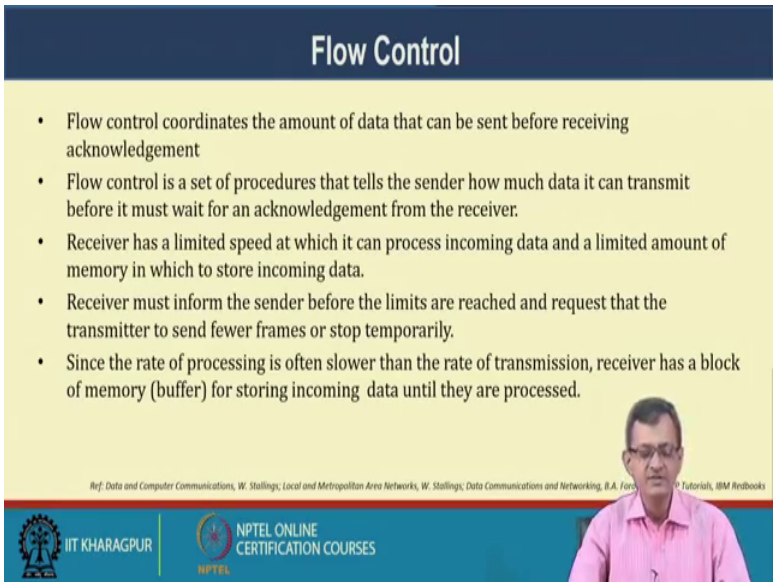
Lecture – 50
Data Link Layer-Flow and Error Control

Hello will continue our discussion on Computer Networks and Internet Protocol, in last few lectures we are discussing on data link layer 2 in the OSI, I also in important layer, I also in TCP IP. So, we have looked into ethernet and different other features of layer 2 and today we will see something on Flow and Error Control.

You will find similarity or between the flow and error control, what you have studied in transport layer or in the transport layer phenomenon, where it in transport layer, it takes care of the end to end process right that process a process in system 1 is process P, in system 1 talking to process Q in system 2 and there is a process to process communication. Where as in this case we have a hub to hub communication right that in layer 2 hub to hub this is system connected to a to a nearby switch.

So, it is that hub connectivity is there, you also and has to you know that that a particular ethernet frame or has a source address, destination MAC address and the payload which is getting from the higher layer right and with other things like error checking, preamble (Refer Time: 01:44) type of things. So, what is important, when we communicate specially these days, when you have switched network switch full duplex networks and things that the to, how this error and flow control mechanism are handle in this scenarios right. So, will take a overall overview of the work flow and error control and as I am again repeating that the overall mechanism is conceptually is similar to that what we have seen in the transport layer, but here at the we are doing at the data link layer.

(Refer Slide Time: 02:30)



So, flow control coordinates amount of the data that can be sent before receiving the acknowledgement right.

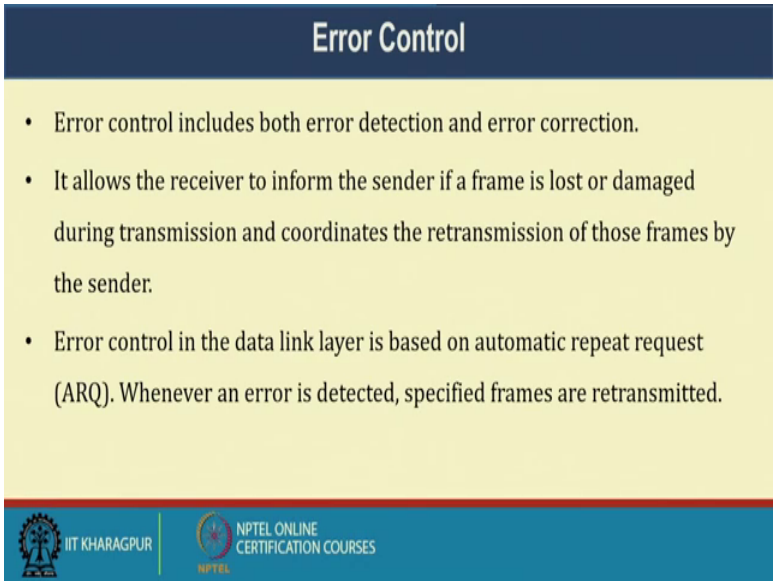
So, what is there? Whenever I am sending a data to somebody right so, after sending I am expecting acknowledgement, the data is received or not to resend. So, what flow control coordinates the amount of how much data, you can send before receiving the acknowledgement type of things, every bitwise byte wise, kilobyte wise or chunk wise and whatever the things right. Flow control is a set of procedure that tells the sender, how much data it can transmit before, it must wait for an acknowledgement from the receiver right.

So, that is a mechanism it says that you say I say, that you sent every packet of so much things or every frame and get a acknowledgement send and explain, what I say that you sent 5 frames wait for the 5 acknowledgement and type of things I say 5 frame is sequence and then wait for the last the maximum acknowledgement thing that all will be there. So, it is say mechanism, it is a procedure, it is a protocol, it is a overall procedure need to be there. Receiver has a limited speed at which, it can process incoming data and a limited amount of memory in which store the incoming data. So, what we see that the receiver it is basic at the receiving in it is a speed at the limited speed (Refer Time: 04:00) it can receive.

It can store some of data type of things receiver must inform the sender, before the limits are reached and request that transmitter to send fewer frames or stop temporarily right. So, if it is overall with the data, it should must sender the sender that 2 thing, it is hungry of the data, it is basically depth of data as the sender to send more or if it is overall in the data, it should say that the stop sending and or at a reduced (Refer Time: 04:33) right. So, this all are has to be through a procedure, which is primarily a flow control mechanisms right.

Since the rate of processing is often slower than, the rate of transmission receiver has block of memory or buffer for storing the incoming data, until their process right. In general the rate of processing with more high speed things are coming up. So, rate of processing the data is often slower than, the rate of transmission. So, usually you are overwhelming with that data. So, receiver has to maintain a buffer or a memory block, where the data can be incoming data can be stored and process like whatever is in the store and forward it store the data process the data and type of things, but there is a that is a limit for the buffer also that, how much you can, it is not a indefinite buffer, you can go on storing indefinite time right. So, that is a restriction on the how much buffer, you can store the data and type of things.

(Refer Slide Time: 05:48)

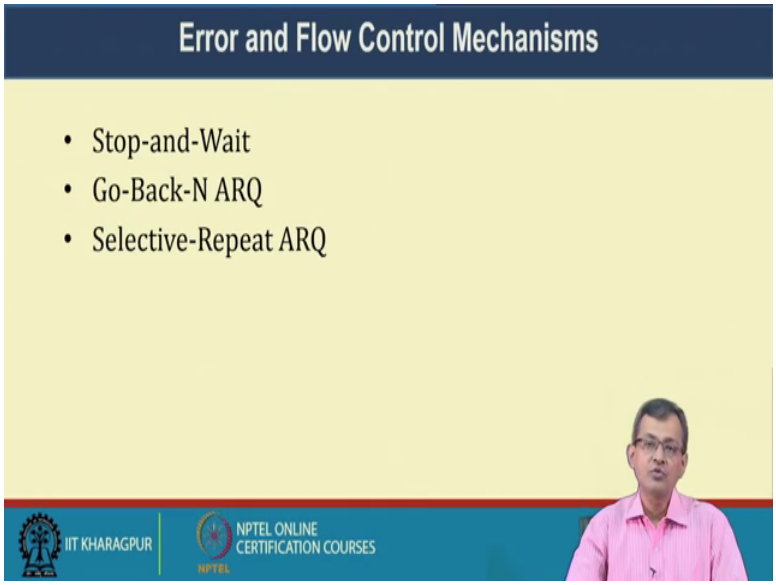


So, on the other hand so, one aspects of a ethernet or layer 2, the data link layer is to have a mechanism for flow control right otherwise, faithful communication at expected

rate etcetera, will be difficult to maintain unless, there is a flow control mechanism which manages the. So, that is that is one important thing, other in aspects of the thing the error control. The data is sent from the receiver sender to the receiver, receiver does not reach it, received it or while sending the acknowledgement, acknowledgement is lost or delayed sometimes it is not lost, it is delay it corrupted and type of things come up into the things. So, that is only important to the how to over all control the error in this layer 2 transmission.

So, error control includes both error detection and error correction right, it allows receiver to inform the sender, if the frame is lost or damaged during the transmission and coordinate the retransmission of those frames by sender right. So, it allows us receiver to inform the sender that whether the frame is lost or damaged during, the transmission and coordinates the retransmission of those frames by the sender, if there is a lost of frame, damaged of the frame that should be retransmission, error control in the data link layer is based on a concept of ARQ or automatic repeat request right, whenever there is a error is detected specified frames are retransmitted right. So, it is a automatic repeat request. So, it is requesting for a automatic repeat of the things. So, that is a mechanisms which is there.

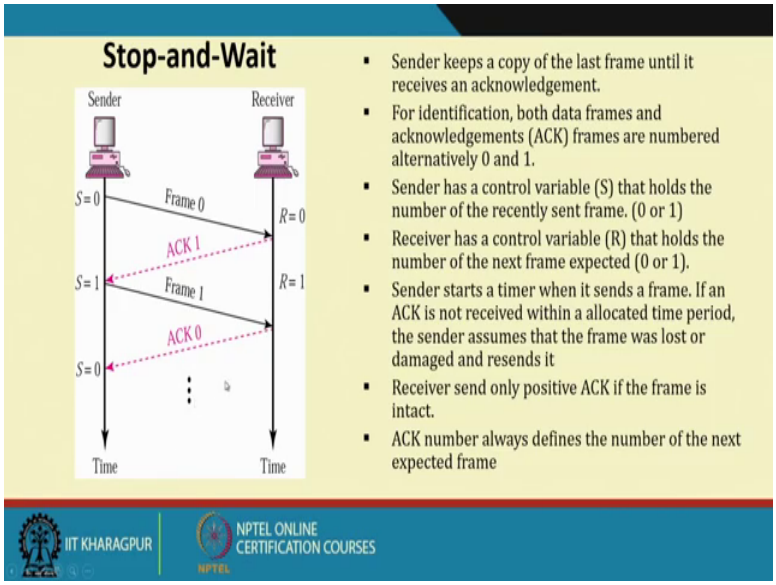
(Refer Slide Time: 07:42)



So, error and flow control mechanism as you might have seen earlier also. So, one is the stop and wait mechanism and Go-Back-N ARQ mechanism selective repeat mechanism.

So, these are the 3 predominant mechanisms, which is followed or which is deployed in the layer 2 of our standard local area network implementations.

(Refer Slide Time: 08:16)



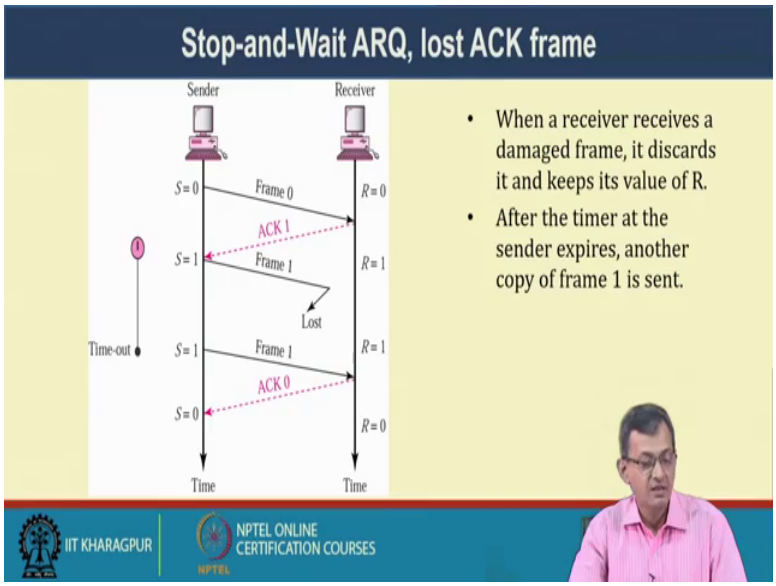
So, in case of stop and wait it is pretty straight forward, sender keeps a copy of the last frame, until it receives a acknowledgement right. So, sender keeps a things for the identification both frames and the acknowledgment are numbered alternatively 0 and 1 so, 01, 01, 01 and type of things right. So, once the frame is sender the 0, acknowledgement comes as a 1 sender frame as 1 acknowledgement comes as 0 and alternatively goes on right.

So, the receiver on receiving the acknowledgement it (Refer Time: 08:56) successfully transmitted. The sender has a control variable S and holds the number of the recently sent frame 0 or 1 right. So, control variable takes care of the things, the receiver as a control variable R and holds the number of the next frame expected 0 or 1 right. Sender start a timer, when it says the frame if an acknowledgement is not received within a allocated time period or within the threshold period or within the timeout period, the sender assume that the frame was lost or damaged and resends it right. Receiver sends only positive acknowledgement, if the frame is intact acknowledgement number always defines the number of the next expected frame right.

So; that means, you see here the acknowledgement 1 missed the next expected frame is 1 acknowledgement, 0 is the expected frame is with the is of the sequence 0 and type of

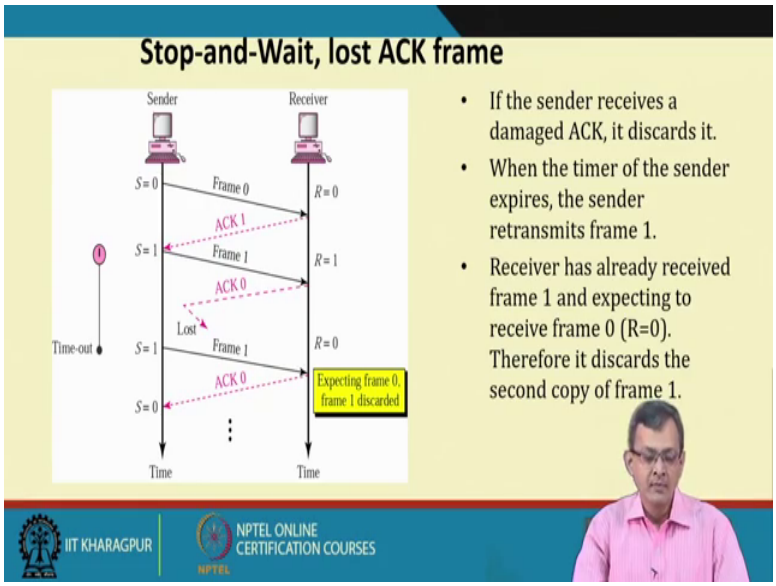
thing. So, this is a very simple mechanism, but it up this allows for this flow an error control rate. So, frame 0 is send acknowledgement send backs 1 on receiving the things it act for the frame 1 is send acknowledgement send as a 0 and so on so forth. So, alternatively 0 1 0 1 things will go.

(Refer Slide Time: 10:28)



Stop and wait ARQ lost acknowledgement frame, there is a there is a acknowledgement frame is lost right. So, when receiver receives a damaged frame, it discards it and keep the value keeps the value of R. In this case, if receive the damaged frame and discards it and keep the R equal to 1, after the timer at the sender expires, another copy of the 1 is sent right. So, it once within the time limit, the sender does not receive the acknowledgement, it will send the another that frame 1 is retransmitted to the receiver means sender to receiver right.

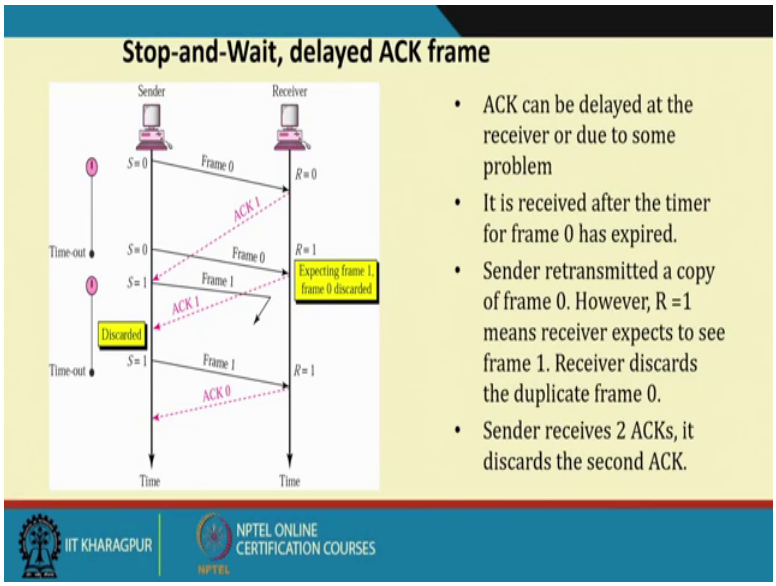
(Refer Slide Time: 11:13)



So, this is a sorry, it is a lost frame right. So here, it is a lost acknowledgement frame. So, frame this is lost acknowledgement frame. So in this case, if the sender receives a damage acknowledgement in discards it right. So, the sender gets acknowledgement, which is a damaged one and it discards it and how it is indentifying? Because you you mean the format we are FCS right frame checks sequence (Refer Time: 11:52) 32 and so on so forth. It can it is able to find out, the there is a error or not, when the timer of the sender expire, the sender retransmit the frame 1 right. Receiver already receives the frame 1 expecting to received frame 0 therefore, discards the second copy of frame 1 right because, it is already received the frame 1 found it faithfully and, but received again frame 1.

So, it discards the second copy because, it is expecting a frame 0 right. So, that way it goes on means, it handles that lost acknowledgement. So, even the acknowledgement is lost in the retransmission phase, it can basically detected.

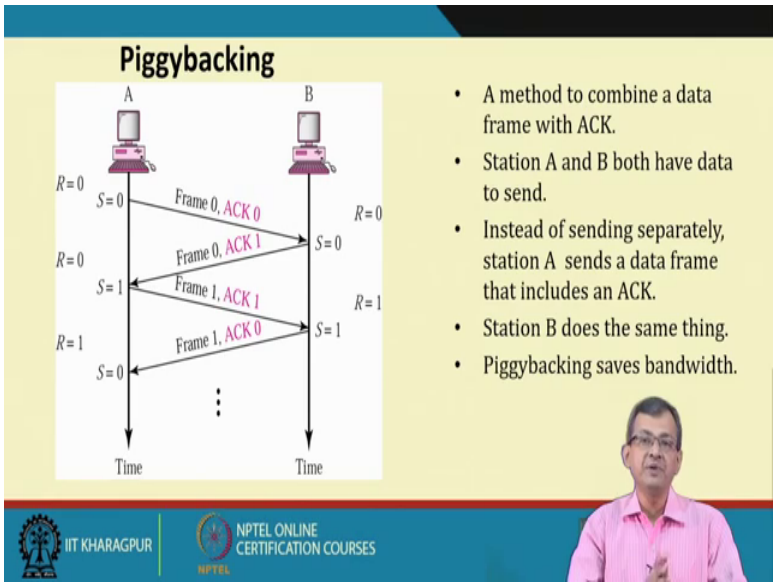
(Refer Slide Time: 12:42)



Stop and wait, delayed acknowledgement frame. So, happen that acknowledgement is same, but it is delayed right acknowledgement can be delayed at the receiver or due to some problem right. It is received after the timer for the frame 0 has expired sender transmitted a copy of the frame 0; however, R equal to 1 means, the receiver expect to see a frame 1 right. So, receiver discards the duplicate frame 0 right. The previously what to was there?

That it has a duplicate 1, because of that acknowledgement problem here the, there is a delayed acknowledgement and the sender things that the frame has lost, it will retransmit the frame again and it discards the frame 0. The frame 0 twice transmitted discards the 0 sender receives 2 acknowledgement, it discards the second acknowledgement if the sender is 2 acknowledgement because, the actual delayed 1 then, the now actual 1 and then it discards the second acknowledgment.

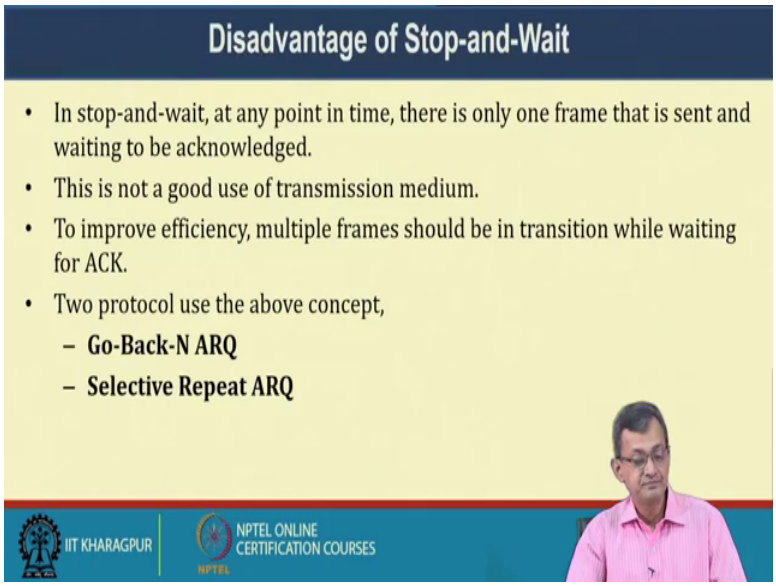
(Refer Slide Time: 13:57)



There is a concept of Piggybacking right. Now because, this both sending the data and acknowledgement are eating away space. So, when we have a both communication, both side communication like station AB, both have data to send then the a method to combine data frame with acknowledgement, the data frame itself carry the acknowledgement.

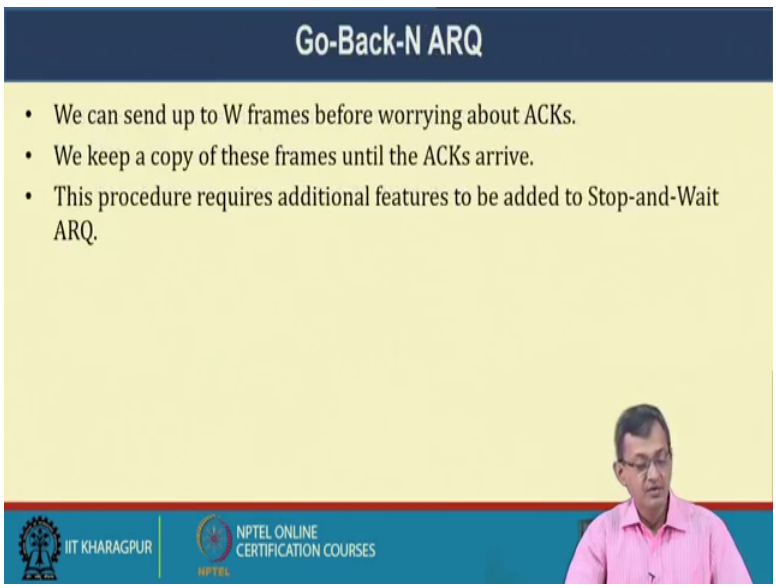
So, the acknowledgement frame Piggyback on the data frame instead of sending separately station a sends the data frame that includes an acknowledgement. Station B does the same thing, Piggybacking saves bandwidth right. So, instant otherwise this acknowledgement frames should have been another channel right; however, small it may be it requires some bandwidth to transmit right. So, this Piggyback on the other means on the data, which are send across the across A and B.

(Refer Slide Time: 14:59)



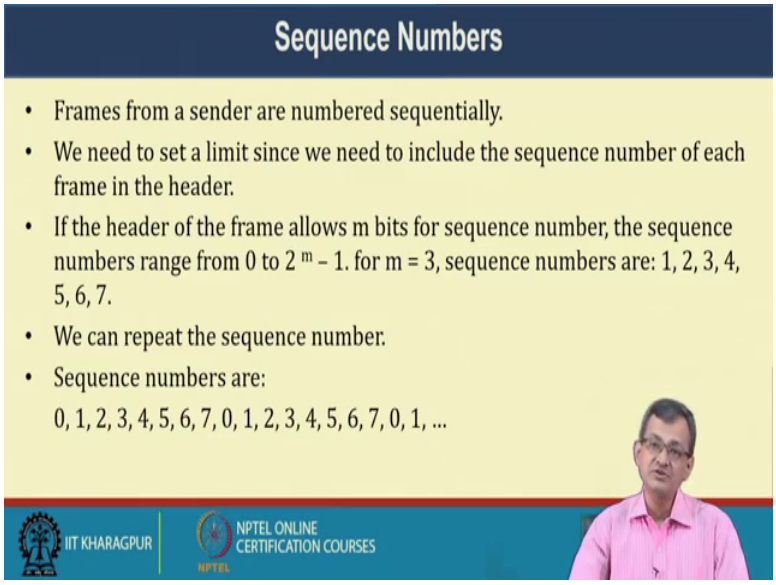
Disadvantage of stop and wait. In stop and wait, any point of time there is only 1 frame that is sent right and waiting to be both the acknowledgement. So, it is sent waiting for the acknowledgement still the timeout and type of things right. This is not a good way of utilising the transmission media so, that is a when the media keeping the media under utilise or not utilised for a longer time is not acceptable. So, there are 2 protocol is one is Go-Back-N, another is Selective Repeat ARQ, Go-Back-N ARQ and Selective Repeat ARQ, which are which are used to take care of these underutilised bandwidth or that the problem.

(Refer Slide Time: 15:50)



So, Go-Back-N ARQ, we can send up to W frame before the worrying about the acknowledgement, send that W frames, we keep a copy of these frames until the acknowledgement receives. This procedure requires additional features to be added in the stop and stop and wait ARQ right. So, which bunch of frames are same in and the copy of kept instead of waiting that every acknowledgement, this the acknowledgement is coming it is goes on reconciliation or what we say that, goes on acknowledging the frames are if the acknowledgement not received and send retransmission required so and so forth.

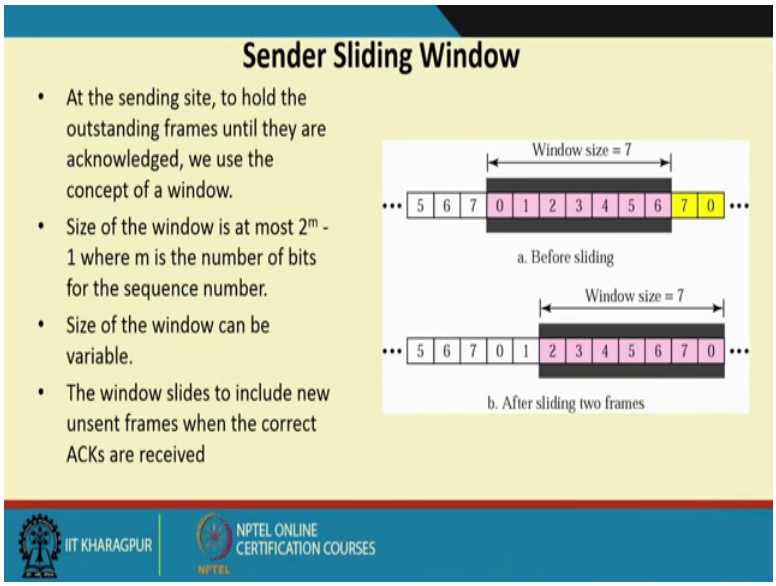
(Refer Slide Time: 16:32)



So, as we bunch of frame, there is not 1 sending and receiving and type of thing. So in this case, what we require is a sequence number right, frames from a sender are numbered sequentially, we need to set the limit since, we need to include the sequence number of each frame in the header, set a limit. So, if the header frame allows m bits for the sequence number.

The sequence number range is 2, the power 0 2, the power m minus 1 right or m equal to 3, sequence number is 1, 2, 3, 4, 5, 6, 7 right or this is the sequence number, we are having we can expect the sequence, we can repeat the sequence number of that those things right. So, it is 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, 4, 5, 6, 7 and so on so forth. So, the sequence number can be repeated on the means on a regular basis.

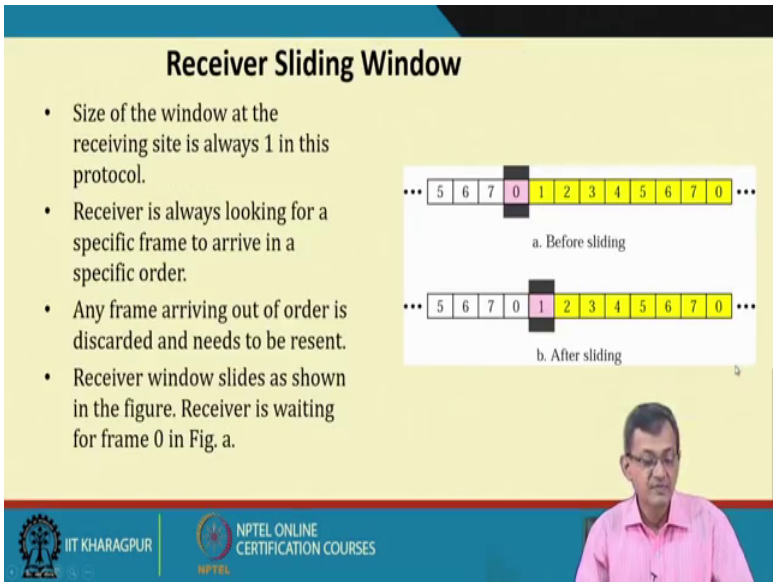
(Refer Slide Time: 17:36)



So, from the sending send the sliding window at the sending a to hold the outstanding frames, until they are acknowledged, we use the concept of a window right, because unless this is acknowledged that has to hold right. We cannot discards those that the sender ends, size of the window is at most 2 the power of m minus 1, where m is the number of bits in the sequence number right.

If there are 3 bit 2 to the power 3 minus 1 type of things size of the window can be variable, how much? That all depends on the (Refer Time: 18:14) type of things the windows slides to include new unsent frames when correct acknowledgement are received right. So, the window slides to include new unsent frames like here, this ones the acknowledgement 0, 1 is it slides to send the next things. So, up to 4 received, it slides again and go on like that.

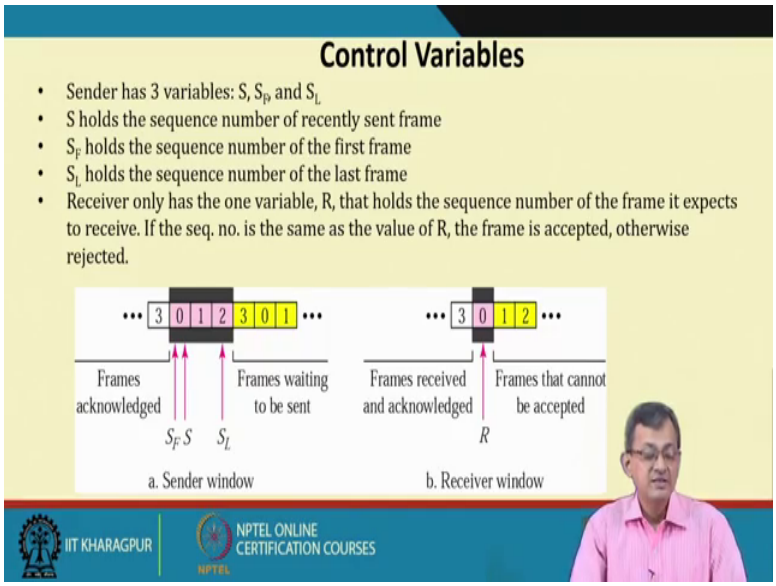
(Refer Slide Time: 18:39)



At the receiving window, size of the receiving window is always 1 for this protocol right. Receiving is always looking for a specific frame to arrive in the specific order. So, whether it is sliding or not at the sender end, receiving is waiting for that specific number of frames, may be in the sequence to wait for the in a specified order right.

Any frame arising out of order is discarded and need to be retransmitted right in a frame arising due to say, monthly path, delay, etcetera. So, will be out of (Refer Time: 19:16). Receiver window slides as. So, this is the receiver window slide shown in the figure, the receiver is waiting for frame 0, in figure 1 and then 1, it is received slides to the waiting for frame 1.

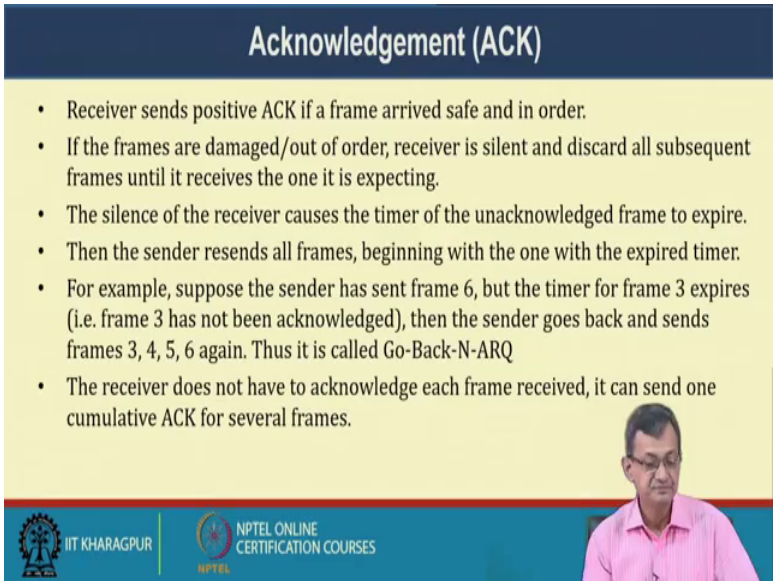
(Refer Slide Time: 19:34)



Control variable, sender has 3 variable S, SF and SL. So, 1 is the end of Markats that is what we say first frame, last frame and S is the currently send frame. So, right receiver has only 1 variable R to hold the sequence number of the frame, it expects to receive. So, it is expecting 0 to receive it will be on that things, if the sequence number is the same as the value of R.

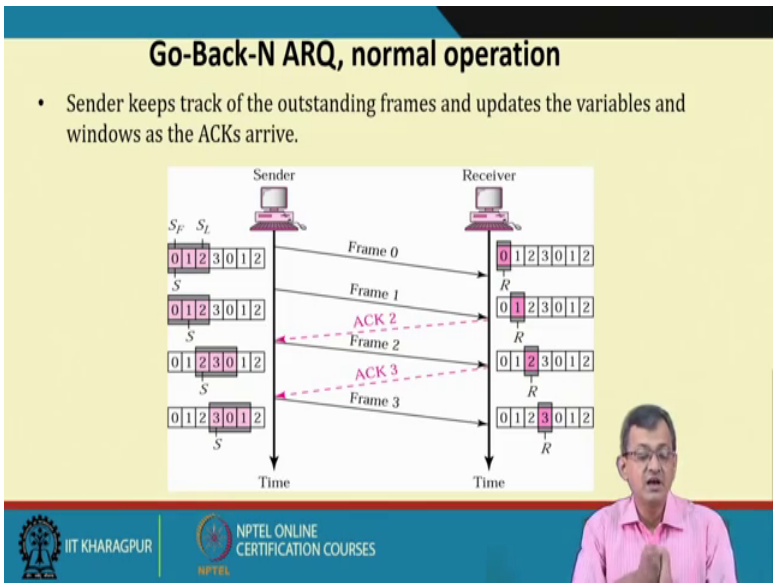
The frame is accepted otherwise rejected. So, the if things received as 0 as it is waiting for the 0, then it is accepted otherwise things are rejected.

(Refer Slide Time: 20:14)



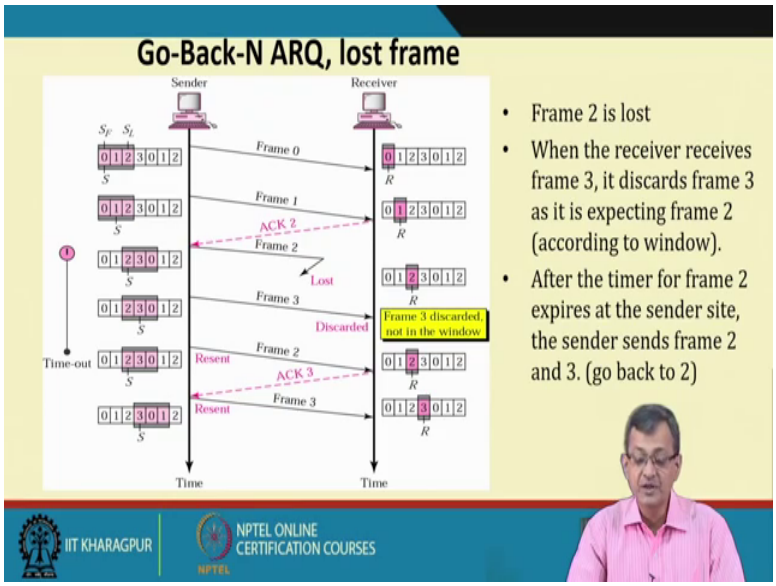
Acknowledgement as we see plays important role right that how the acknowledgement is there receiver sends positive ACK if a frame arrived safe and in a order; that means, it is correct and in order. If the frames are damaged out of order receiver silent and discards all subsequent frames until it receives the one is expecting. So, it is instead of it is expecting 1 instead of 1, it is receiving 2 3 etcetera, it discards all frames right then the sender resends all the frames beginning with the 1, which has expired timer for example, suppose the sender has frame 6, but the timer for the frame 3 expires then, the sender go back and sends the frame 3, 4, 5, 6, this is called Go-Back-N ARQ. The receiver does not have to acknowledge each frame received, it can send 1 accumulate accumulative acknowledgement for several frames that is also possible.

(Refer Slide Time: 21:12)



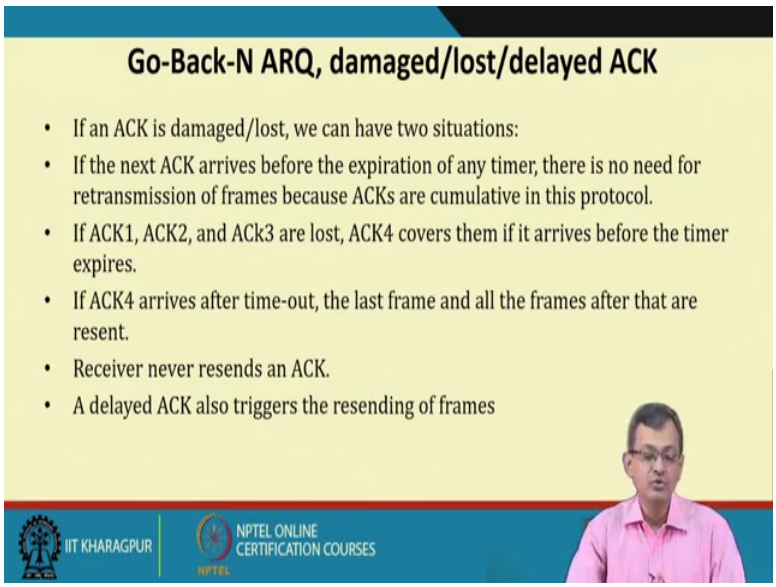
So, this is the Go-Back-N ARQ, sender keeps track of the outstanding frames that are an updates the variables and windows at the acknowledge arrives right.

(Refer Slide Time: 21:33)



And in case of a lost frame. Frame say in this case, frame 2 is lost, when the receiver receives frame 3 discards frame 3, it is expecting 2 as we are discussing. So, it has expecting, 2 here and receive 3 instead of 2 discards. The timer from the 2 expire and the sender side and sender sends frame 2 and 3, go back to N, that is go back to 2, in this case.

(Refer Slide Time: 22:01)

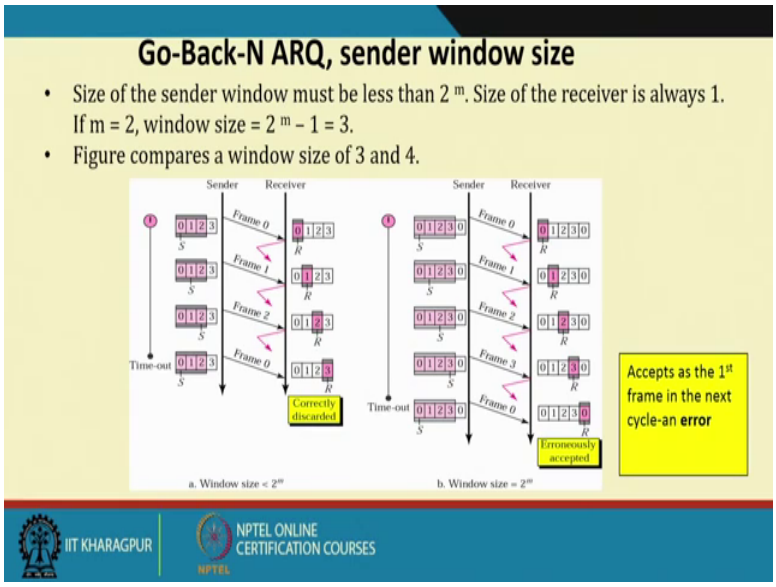


Go back to Go-Back-N ARQ damage, lost, delayed acknowledgement. If the acknowledgement is damaged or lost, we can have 2 situations, if the next

acknowledgement arrives before the expiry of any timer, there is no need for retransmission of the frames because, the acknowledgement are cumulative in this protocol. So, it is a cumulative. So, that is the next acknowledgement there is no (Refer Time: 22:30) of retransmission if a AKC 1 sorry, ACK 1, ACK 2, ACK 3 are lost. ACK 4 covers them if it arrives before the timer expires, then it say it say cumulative thing. If ACK 4 arrives after the time out, the last frame of all the frames after that are resent right, receiver never resends an acknowledgement. So, receiver never resends an acknowledgement, it is only 1, it is sends a delayed acknowledgement also triggers the resending.

So, a delayed acknowledgement is also in the case of a is like a damage or lost and it triggers a resending of the frames.

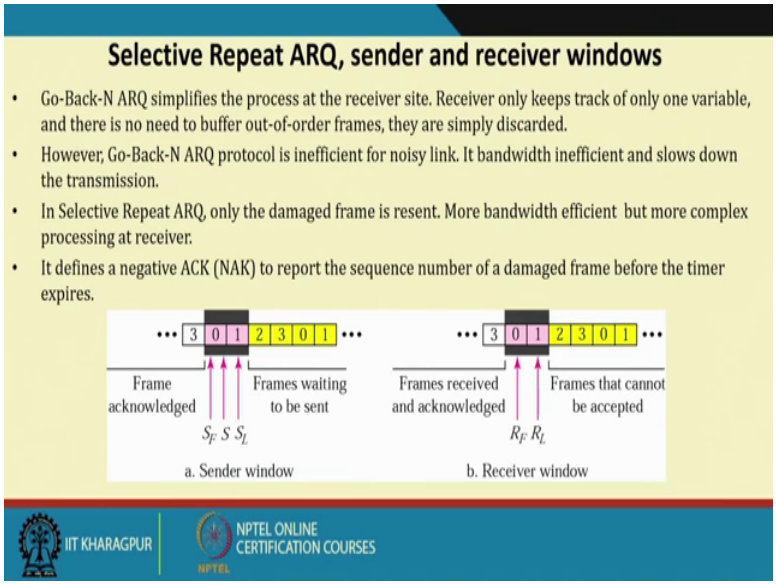
(Refer Slide Time: 23:12)



So, on the sender side Go Back N ARQ. The size of the window must be less than 2 power m, size of the receiver is always 1, if is equal to 2 the size is 2 to the power 2 minus 1 is 3. So, figure compares the window size of 3 and 4 that how things will be there accepts as a first frame in the next cycle and error and so, it goes on looking at the things right. So though, there is a lot of frames are lost, it has got erroneously got the things erroneously 0 and it is expecting this 0 and mistaking to the 0 and which is erroneously expected, where is in this case, if there is a loss, it is it is expecting 2 and but

at the 3 correctly discarded. So, is not received a 0. So, it is correctly discarded right. So, it is also that window size plays role in this sort of scenarios.

(Refer Slide Time: 24:26)



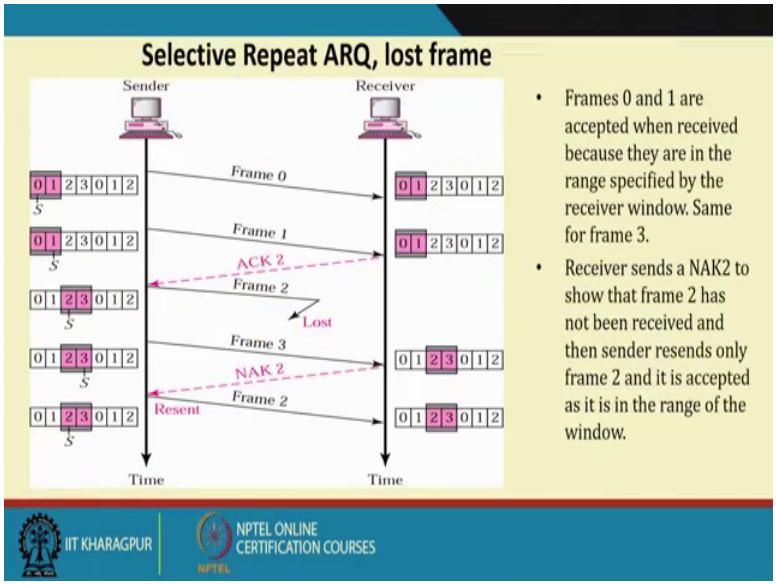
Selective ARQ sender and receiver windows, this is another ARQ mechanisms. So, Go-Back-N ARQ simplifies process of the receiver size right, receiver only keeps track of only 1 variable right and there is no need of buffer out of order frames and simply discard the things, if anything out of the frame coming, in case of a Go-Back-N ARQ.

Then you just simply discard the thing. So, was the thing right; however, in selective repeat ARQ only damage frames are resent more bandwidth efficient, but more complex at the processing in the it means selective repeat; that means, you are not discarding, in case of go back N in for finding 1 out of water, discard everything after that, but in selective repeat you are not discarding all you are only discarding, which are erroneous or damage frames and type of things right you are you. So, selectively requesting resend from the sender for the selective frames. It sends, it defines a negative acknowledgement NAK to report the sequence number of a damage frame, before the timer expires. So, it sends a negative acknowledgement or NAK right.

So here, instead of in the case of Go-Back-N, what we have seen there are this is SF, SL and S was there and whereas, here we R F and R L that cannot be accepted frame received and acknowledged right out of that things, which are received and

acknowledged here frames acknowledged, frames awaiting acknowledgement is after that right.

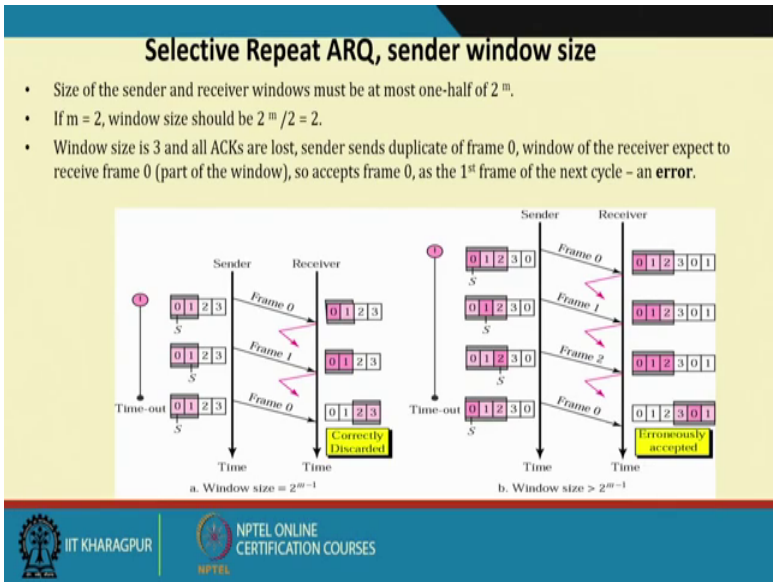
(Refer Slide Time: 26:33)



So, in selective repeat ARQ, lost frame 0 and 1 are accepted, when the received accepted when receive because, they are in the range specified by the receiver window of the frame (Refer Time: 26:47).

So, it is within the receiver window size. So, it is 0 and 1 are accepted, when received because they are in the range of the (Refer Time: 26:58) right. Receiver sends a receiver sends a NAK2 to show that the frame 2 has not been received right and sender the resends only frame 2 and it accept and it is accepted as it is the range of the window right. So this, a range of the window matters much. So, if it is within the range of the window. So and then there you receive a NAK of request repeat or selective repeat of a particular frame, then it is resend by the sender.

(Refer Slide Time: 27:35)



So, again selective repeat ARQ is the sender windows, window again plays a important role, size of the sender and receiver window must be at most half that 2 to the power m; that means, m is equal to 2. So, size will be 2 to the power 2 by 2 2 so window size is 3 an all acknowledgement as lost sender sends duplicate frame 0 right like in other case, also if it is window size is greater than 2 to their m minus 1; that means, to your m divided by 2 then, I can receive accept this 1 though, it was damage, I accept this erroneously accepted, where as if it is we need that thing, that it is correctly discarded by the receiver.

So, window size 3, all acknowledgement has lost. Sender sends duplicate frames 0 windows of the receiver expect receive frame part of the windows accept the 0 and the first frame of the next cycle. So, it is a error. So, it is accepted in a erroneous manner right. So, what we see here, in this sort of flow and error control mechanisms. So, the sender receiver with help of acknowledgement, in case of the first 1 stop and wait it is everything is acknowledged, every frame to individually acknowledged, but in go back in what we have seen that, the receiver has only 1 pointer, where as a bunch of thing, bunch of frames can be send from the receiver from the sender end. And it also send a cumulative acknowledgement has a option of sending cumulative acknowledgement in repeat ARQ selective repeat ARQ.

So, in case of go back end if there is a error everything after that need to retransmitted where as the mechanism is the NAKs mechanism has a better utilization of the bandwidth when is a repeat ARQ, they need it sends only those selective frames which has damaged or erroneous things like that right. And in both the cases what will ask 2 cases, what your seen this window size place the important role right, otherwise you may erroneously accept something right, if the window size is not properly maintain. So, that is important for things, this flow and error control mechanism over and above tries to have provide a better utilization or better flow mechanisms within the things.

So, you have a better utilization of the bandwidth or provide a better quality of service for this mechanisms right. So, this is the overall flow and error control mechanism. So, with this we look we conclude the our basic flow and error control mechanism in the data to data line clear and also we see that this provides, say this helps us in providing a better utilization of the bandwidth right. There is 2 issues are there 1 is the bandwidth is not available right, you have a 10 MBPS and more data to be you have a higher bandwidth, but due to in proper utilization, you may not be exploiting that in a proper.

So, in subsequent lecture will see other aspect of the things, there are several data, line clear or switching aspect those things. We will be looking at which also which are some of things in also have linkage with higher level network layer phenomenon. So, that will sub in the subsequent lecture or on this particular series of lecture will be looking at. So, let us conclude today.

Thank you.