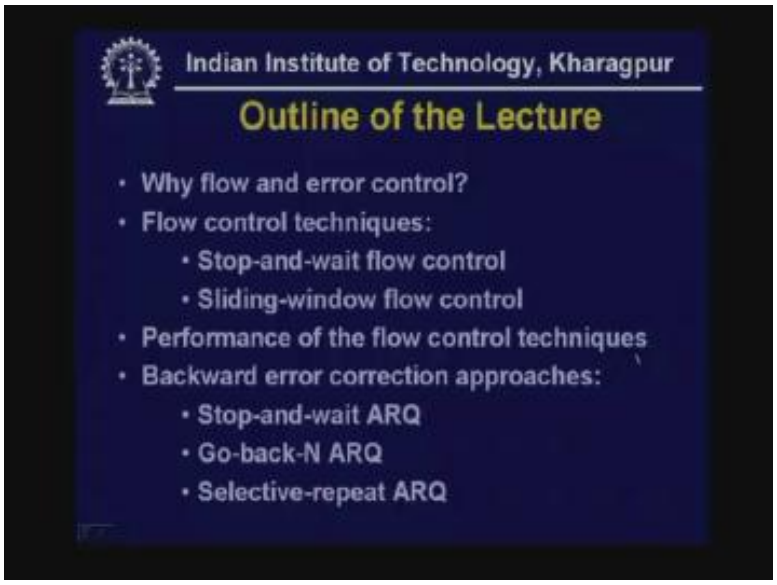
Data Communications
Department of Computer Science & Engineering
Prof. A. Pal
Indian Institute of Technology, Kharagpur
Flow and Error Control
Lecture - 16

Hello viewers welcome to today's lecture on flow and error control. In the last lecture we have discussed about various error detection techniques and also we have discussed about a technique for forward error correction k using hamming code. Now, for successful data communication it is necessary to use some techniques so that data communication can be performed in a reliable and efficient manner. So in these contexts two techniques flow control and error control are very important. In this lecture we shall discuss about the flow and error control techniques. Here is the outline of today's lecture.

(Refer Slide Time: 01:48)

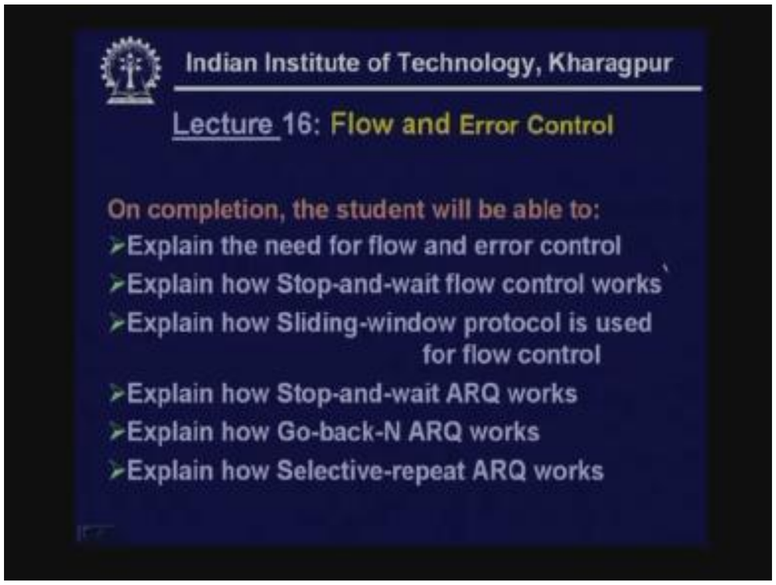

Indian Institute of Technology, Kharagpur

Outline of the Lecture

- Why flow and error control?
- Flow control techniques:
  - Stop-and-wait flow control
  - Sliding-window flow control
- Performance of the flow control techniques
- Backward error correction approaches:
  - Stop-and-wait ARQ
  - Go-back-N ARQ
  - Selective-repeat ARQ

First we will discuss why flow and error control is necessary then we will discuss the two important flow control techniques stop-and-wait flow control and sliding-window flow control and we shall compare the performance of these two flow controls techniques namely stop-and-wait flow control and sliding-window flow control. Then we shall see how we can perform the backward error correction using three techniques; stop-and-wait ARQ, go-back-N ARQ and Selective-Repeat ARQ. So, we shall consider these techniques one after the other.

(Refer Slide Time: 2:38)



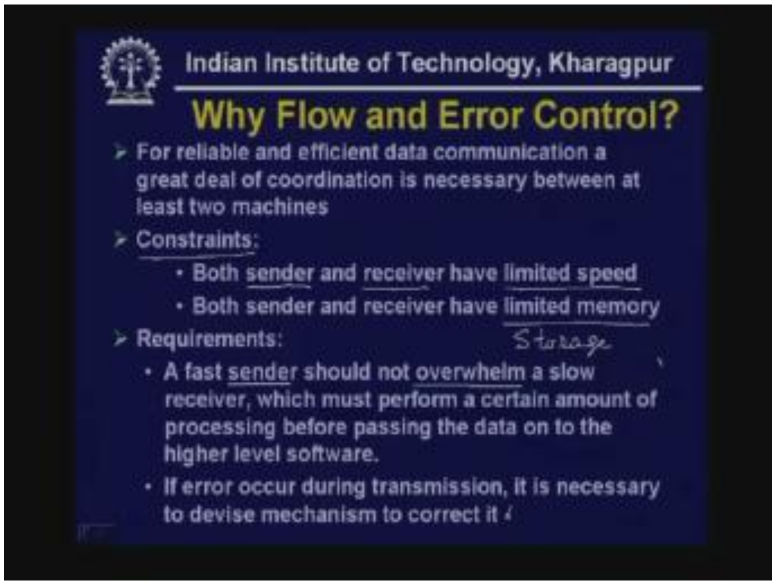
Indian Institute of Technology, Kharagpur

Lecture 16: Flow and Error Control

On completion, the student will be able to:
- Explain the need for flow and error control
- Explain how Stop-and-wait flow control works
- Explain how Sliding-window protocol is used for flow control
- Explain how Stop-and-wait ARQ works
- Explain how Go-back-N ARQ works
- Explain how Selective-repeat ARQ works

And on completion of this lecture the student will be able to explain the need for flow and error control, they will be able to explain the stop and flow control techniques how it works, they will be able to explain how sliding-window protocol is used for flow control, then they will be able to explain how stop-and-wait ARQ works, they will be able to explain how Go-back-N ARQ works and also they will be able to explain how Selective-Repeat ARQ works. They will also be able to compare the relative performance of these three techniques. First let us focus on why flow and error control.

(Refer Slide Time: 03:25)



Indian Institute of Technology, Kharagpur

Why Flow and Error Control?
- For reliable and efficient data communication a great deal of coordination is necessary between at least two machines
- Constraints:
  - Both sender and receiver have limited speed
  - Both sender and receiver have limited memory Storage
- Requirements:
  - A fast sender should not overwhelm a slow receiver, which must perform a certain amount of processing before passing the data on to the higher level software.
  - If error occur during transmission, it is necessary to devise mechanism to correct it

As I mentioned for successful data communication we have to use certain techniques. Now we know that for data communication we require at two devices of two machines; one is sender another one is receiver and we require a great deal of coordination between these two devices or machines for reliable and efficient data communication.

Let us see what are the constraints involved in it. First of all one constraint is both sender and receiver have limited speed. That means they will require some time to receive data to process data and to store data. So, because of some limited speed it will involve some time.

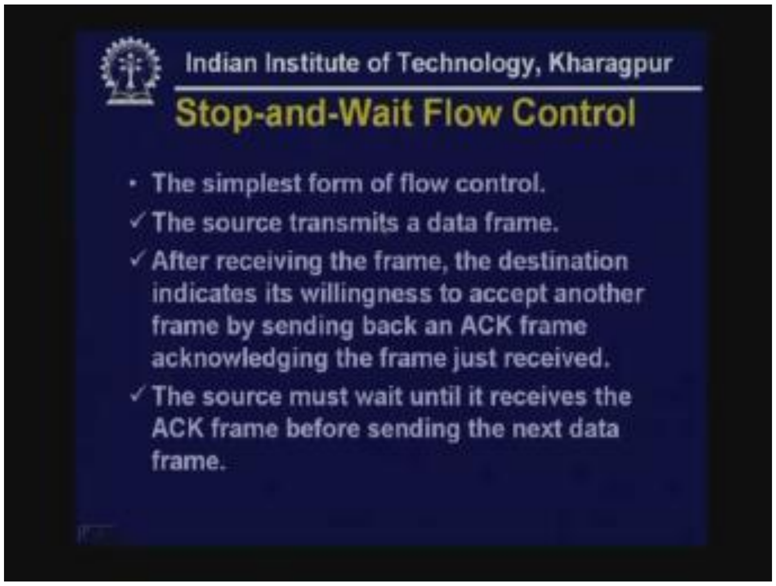
Similarly for both sender and receiver have limited memory for storage capability and usually it is known as buffer because whenever data communication takes place usually the data that is being received is temporarily stored in memory called buffer so that capacity is limited. Thus under these constraints the requirements are; a fast sender should not overwhelm a slow receiver which must perform a certain amount of processing before passing the data on to the higher level software. That means after the data is received the receiver has to do some processing before it can pass on to next higher level software or store it in a permanent memory.

Now we know that different kinds of machines. For example, a surfer is usually a high end machine which can perform processing at higher speed. On the other hand the client or receiver can be a little low end machine so obviously its speed or processing capability is lower compared to the server. In such a situation the sender should not….. at such a rate that the receiver is overwhelmed so you have to protect the receiver from being overwhelmed and for that purpose the sender must send data in a very controlled way because unless it is controlled in a control way there will be overflow from the buffer or information will be lost therefore that controlled way is known as flow control. That means the receiver will send some acknowledgement before the sender sends some data. By using that kind of coordination overwhelming can be overcome and data communication will take place successfully without any loss of information.

Another information is that if error occurs during transmission it is necessary to devise a mechanism to correct it. We have already discussed the error detection techniques and also one forward error correction technique using hamming code. But commonly forward error correction technique is not used. The most commonly used technique is that the receiver will check the received data and if an error is found it sends information to the sender that error has occurred in the received data then the sender will retransmit the data again. This technique is known as backward error correction or error control technique.

Hence for reliable and efficient communication we have to perform flow and error control. So in this lecture we shall discuss about these things. First we shall focus on the simplest flow control the stop-and-wait flow control.

(Refer Slide Time: 07:40)



Indian Institute of Technology, Kharagpur

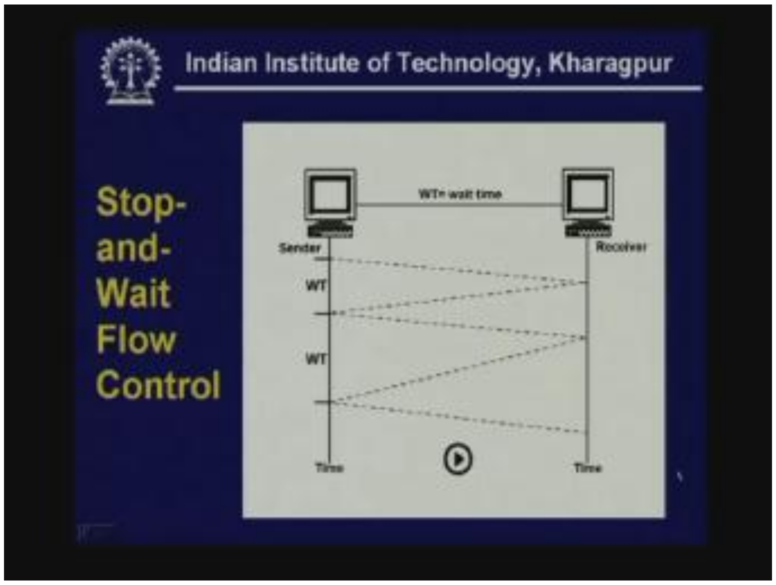
Stop-and-Wait Flow Control

- The simplest form of flow control.
- ✓ The source transmits a data frame.
- ✓ After receiving the frame, the destination indicates its willingness to accept another frame by sending back an ACK frame acknowledging the frame just received.
- ✓ The source must wait until it receives the ACK frame before sending the next data frame.

Although this flow control and error control are performed in an integrated manner but for the sake of understanding we are considering them one after the other. But usually they are performed in an integrated manner.

In this stop-and-wait flow control the source transmits a data frame. That means sender transmits a data frame and after receiving the frame the destination that means the receiving end indicates its willingness to accept another frame by sending back an acknowledging frame. So the receiver after receiving that frame will send an acknowledgement and it will indicate that the frame has been received already so another frame can be accepted as it is ready for receiving another frame. The source must wait until it receives the acknowledgement frame before sending the next data frame. This is the basic idea behind the stop-and-wait flow control. Let us see how it really works with the help of this animation.

(Refer Slide Time: 08:58)


Indian Institute of Technology, Kharagpur

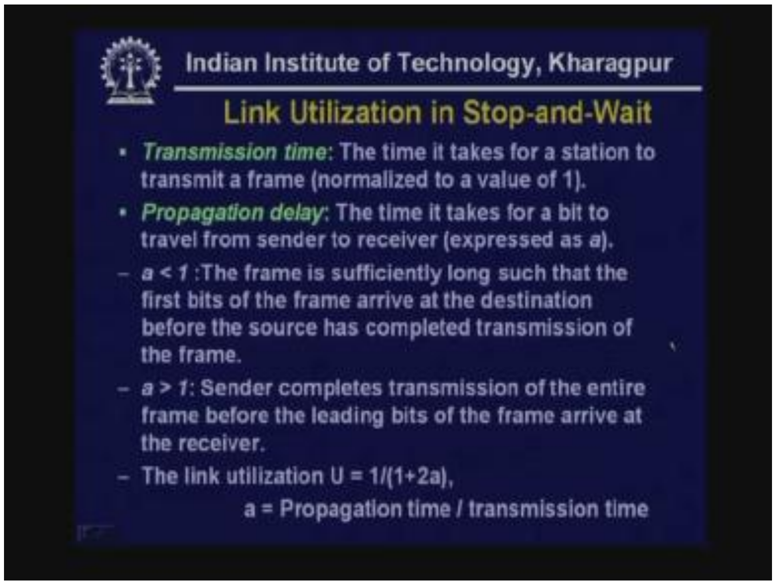
Stop-and-Wait Flow Control

So here is the sender side and here is the receiver side and a frame is ready for transmission and it is going to be sent by the sender. now it is taking some time to reach the receiver known as the propagation time and after receiving the frame the receiver will take some time to do some processing then it will send an acknowledgement and the acknowledgement will reach the sender and after it is received by sender the sender will then send another frame. Hence there is a waiting time involved before another frame can be sent.

Similarly, another frame is received by the receiver and it will send another acknowledgement and after receiving that acknowledgement again the sender will send another frame as you can see from this animation. So here as you can see (Refer Slide Time: 9:55) between two frames there is a waiting time or wait time involved so some time is wasted before another frame can be sent. This ensures that the receiver is not overwhelmed. This is the basic idea behind stop-and-wait flow control.

Let us see the performance of this stop-and-wait flow control technique. For that purpose we have to understand about two important parameters. One is known as transmission time.

(Refer Slide Time: 10:33)



Indian Institute of Technology, Kharagpur

Link Utilization in Stop-and-Wait

- *Transmission time*: The time it takes for a station to transmit a frame (normalized to a value of 1).
- *Propagation delay*: The time it takes for a bit to travel from sender to receiver (expressed as *a*).
  - *a* < *1* :The frame is sufficiently long such that the first bits of the frame arrive at the destination before the source has completed transmission of the frame.
  - *a* > *1*: Sender completes transmission of the entire frame before the leading bits of the frame arrive at the receiver.
  - The link utilization U = 1/(1+2a),
    a = Propagation time / transmission time

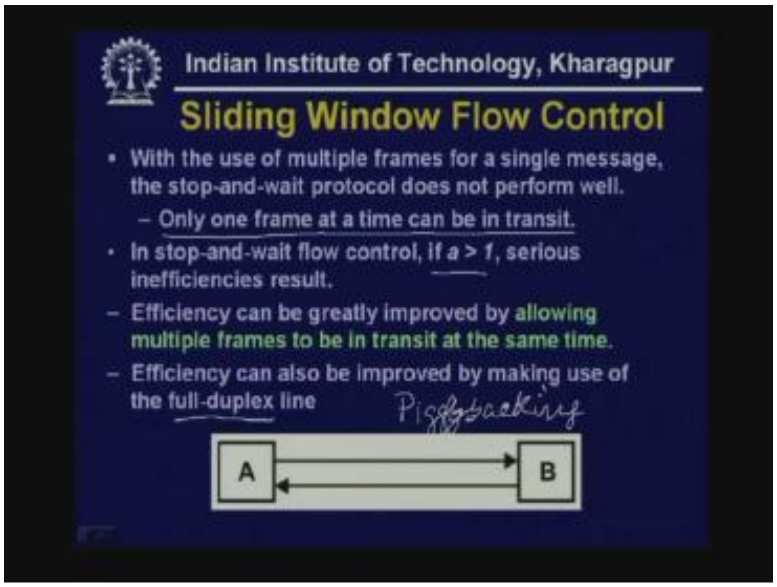
The transmission time is the time it takes for a station to transmit a frame. The time for transmission of a frame will depend on the length of the frame and also it will depend on the rate at which the transmission is taking place. Let us assume that the size of the frame is 10 to the power 3 bytes that means 1 kilo byte and the rate at which it is being sent is 10 Mbps. So when it is sent at this rate obviously the transmission time will be 10 to the 3, 10 to the power 6 which is 1
millisecond.

Now let us come to the propagation time. The propagation time is the time it takes for a bit to travel from sender to receiver and this time is obviously dependent on the distance and the speed of the electromagnetic wave. Therefore longer the distance more will be the propagation time. For example, whenever we use satellite communication then the signal goes from the base station to the satellite and back to the ground station and as we know it take quarter of a second and whenever we are using say fiber optic communication then length of the cable can be very long and in such a situation propagation time can be long. And the ratio between the two is known as ‘a’. So ‘a’ is the ratio between the propagation time by the transmission. Usually transmission time is normalized to the value 1. So, whenever a is less than one the frame is sufficiently long such that the first bit of the frame arrive at the destination before the source has completed transmission of the frame. That means here is the sender (Refer Slide Time: 12:44) and here is the receiver so it has started the transmission.

Therefore in such a situation you will see that before the first bit reaches the destination the transmission will not get complete. That means in this case transmission line is longer than the propagation time. On other hand whenever ‘a’ is greater than 1 in such a case what will happen is, this is the sender’s side and this is the receiver side so before it reaches the other end the transmission is over. For example, the scenario will be somewhat like this (Refer Slide Time: 13:22). That means this is the frame and before this end has reached the receiver the transmission has been completed and some time has already passed. This will be case whenever ‘a’ is greater than 1.

And depending on this value of ‘a’ which is the propagation time by transmission time it can be shown that the utilization of the link is equal to 1/1 plus 2a. So you see that higher the value of ‘a’ utilization will be poor on the other hand smaller the value of ‘a’ utilization will be better. Hence we can see that whenever we are communicating over a long distance then there is a possibility of I inefficient utilization in case of stop-and-wait flow control. On the other hand when propagation time is small then the utilization will be better. This problem can be overcome by using sliding-window protocol.

(Refer Slide Time: 14:18)


Indian Institute of Technology, Kharagpur

Sliding Window Flow Control

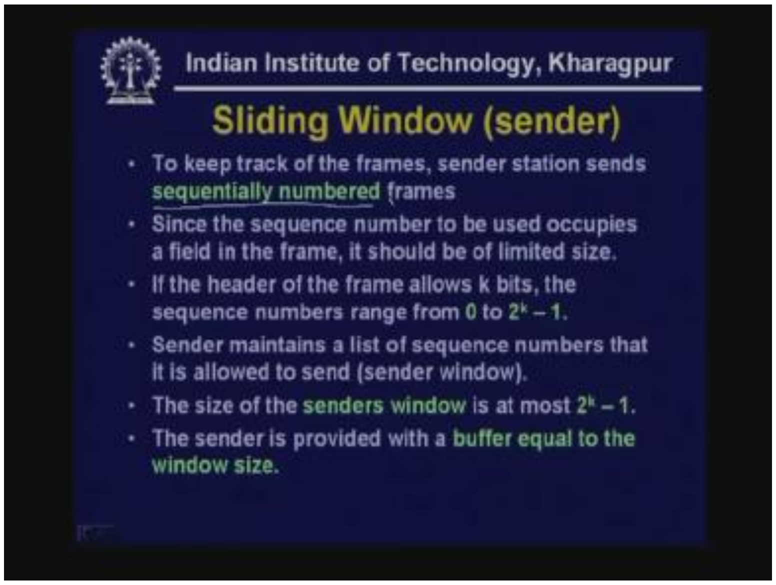
- With the use of multiple frames for a single message, the stop-and-wait protocol does not perform well.
  - Only one frame at a time can be in transit.
- In stop-and-wait flow control, if $a > 1$, serious inefficiencies result.
  - Efficiency can be greatly improved by allowing multiple frames to be in transit at the same time.
  - Efficiency can also be improved by making use of the full-duplex line

The main limitation of stop-and-wait protocol is that at a time only one frame is in transit. Normally a single message is divided into a number of frames. Therefore whenever short frames are sent the transmission line is small and that is used so that error does not occur and for various other reason. So in such a case the stop-and-wait flow control will not be efficient. Therefore when we use multiple frames for single message as the normal scenario is then the stop-and-wait protocol does not perform well. The main reason is that here only one frame at a time can be in transit and that is being overcome in sliding-window protocol as we shall see.

As you have see when ‘a’ is greater than 1 then serious inefficiencies would result in stop-and-wait protocol. Efficiency can be greatly improved by allowing multiple frames to be in transit at the same time. That means before an acknowledgement is received from another end if multiple frames can be allowed to be sent then inefficiency can be improved and that is precisely what is done in case of sliding-window flow control. Moreover efficiency can be further improved by making use of the full-duplex line. As we shall see acknowledgement can be sent from the receiver side as part of information frame known as piggybacking.

Therefore piggybacking technique is used which will improve the efficiency of transmission. Let us see how this sliding-window protocol works.

(Refer Slide Time: 16:38)



Indian Institute of Technology, Kharagpur

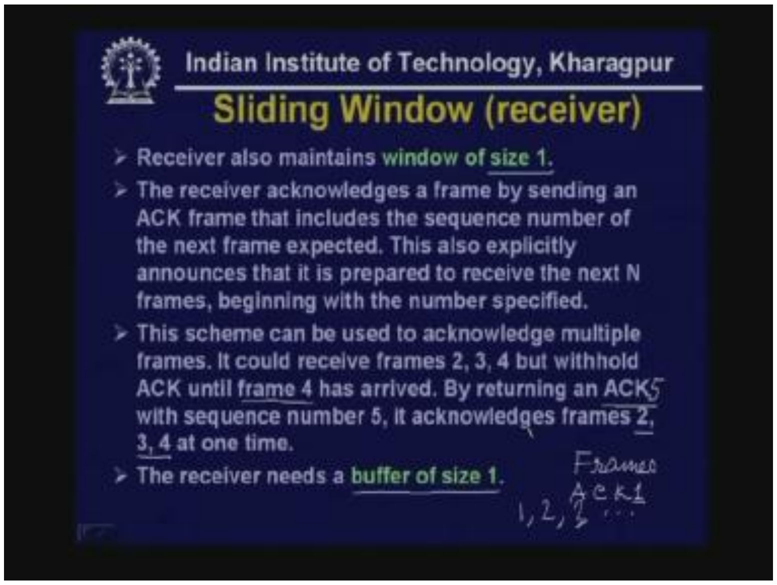
Sliding Window (sender)

- To keep track of the frames, sender station sends sequentially numbered frames
- Since the sequence number to be used occupies a field in the frame, it should be of limited size.
- If the header of the frame allows k bits, the sequence numbers range from 0 to $2^k - 1$.
- Sender maintains a list of sequence numbers that it is allowed to send (sender window).
- The size of the senders window is at most $2^k - 1$.
- The sender is provided with a buffer equal to the window size.

Now, first of all to keep track of frames we are sending multiple frames. Obviously unless the frames and number it will be difficult to keep track which frame has received the destination and which frame has not reached the destination. So the frames are sequentially numbered and they are sent. And since the sequence numbers to be used occupies a field in the frame it should be of limited size.

Obviously the sequence number should be part of the frame. that sequence number that is embedded will be one of the fields of the frame and obviously the number of bits that is used for representing the sequence number will be an overhead and that’s why there is some restriction on the number of bits that can be used. If the header of the frame allows k bits then the sequence number range is from 0 to 2 power k. That means there is a possibility of having 2 to the power k numbers with k bits whenever you are using k bits for representing the sequence number.

(Refer Slide Time: 16:32)


Indian Institute of Technology, Kharagpur

Sliding Window (receiver)

- Receiver also maintains window of size 1.
- The receiver acknowledges a frame by sending an ACK frame that includes the sequence number of the next frame expected. This also explicitly announces that it is prepared to receive the next N frames, beginning with the number specified.
- This scheme can be used to acknowledge multiple frames. It could receive frames 2, 3, 4 but withhold ACK until frame 4 has arrived. By returning an ACK5 with sequence number 5, it acknowledges frames 2, 3, 4 at one time.
- The receiver needs a buffer of size 1.

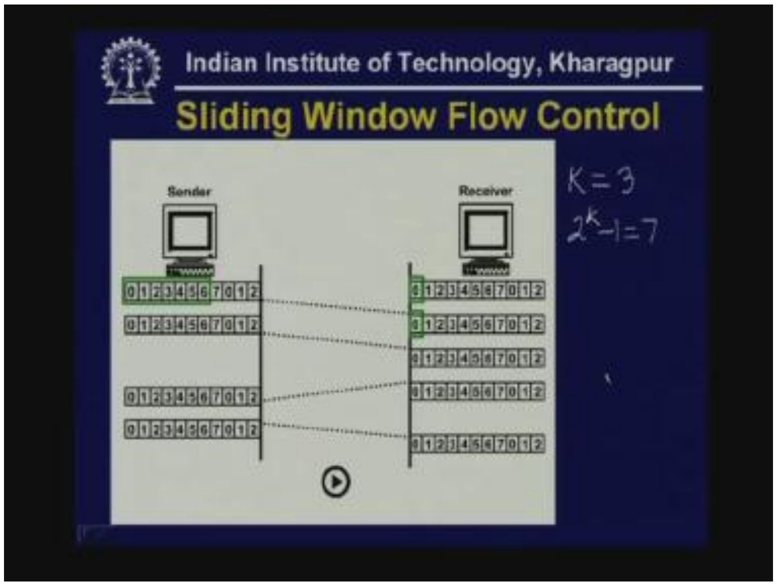
Sender maintains a list of sequence numbers that it is allowed to be sent. That means a window is maintained at the sender's end and it is essentially the number of frames that can be sent say starting with 0 and then if you are using 7 bit it can be up to 0, 1 to 7. You will see that the window size is usually not 2 to the power k but 2 to the power k minus 1 less than this. That means it has to be 1 to 6. That means this is the range of frames which can be sent from the receiver without receiving an acknowledgement. Then the sender is provided with a buffer equal to the window size. That means the sender is having a buffer to store all these frames and then they are sending one after the other. Similarly at the receiving end a window is also maintained but the size of the window is equal to 1.

The receiver acknowledges a frame by sending an acknowledgement frame that includes the sequence number of the next frame expected. That means suppose frame 0 has been received by the receiver then it will send an acknowledgement like this ACK1 that means it is a frame that it is expecting next. So in this way this explicitly announces that it is prepared to receive the next N frames beginning with the number specified. That means with this number 1 like numbers 1, 2, 3 in this way starting with 1 to n minus 1 up to 2 to the power k minus 1 frames it can receive one after the other so this readiness is conveyed by the receiver. This scheme can be used to acknowledge multiple frames.

it is not necessary that for each frame received by the receiver it has to send an acknowledgement. It is possible to send a single acknowledgement for a number of frames. How it can be done let us see?

Suppose it has received three frames two three and four but it holds acknowledgement until frame 4 has arrived now by returning an acknowledgement like ACK5 with sequence number 5 it acknowledges frame 2, 3 and 4 at one time. That means while sending a single acknowledgement ACK5 it serves the purpose of acknowledgement for three frames namely 2, 3 and 4 and obviously the sender will not transmit the frame 5 and in the receiver as I said since the window size is 1 it will require a buffer of size only 1 because at a time it is receiving one frame and then passing it on to the next higher level. Let us see how it actually works.

(Refer Slide Time: 20:55)


Indian Institute of Technology, Kharagpur
Sliding Window Flow Control
Sender
Receiver
K=3
$2^k-1=7$

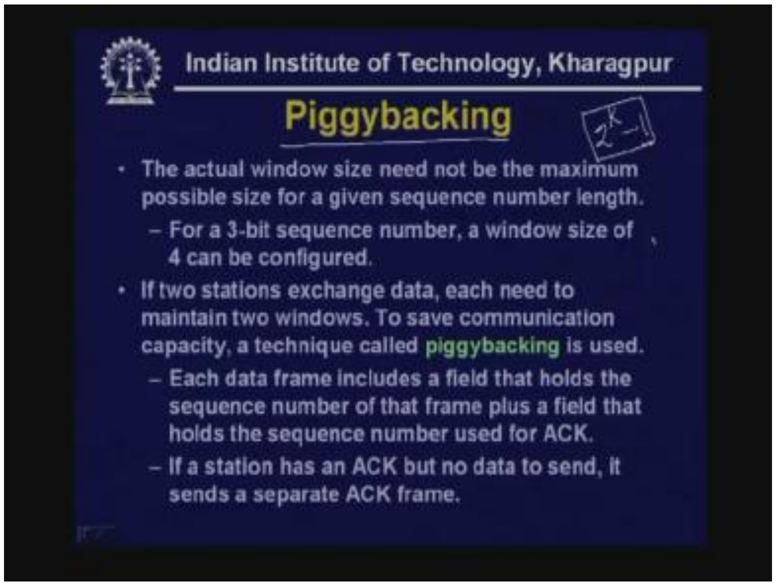
So as we can see here we are using k is equal to 3 so our window size is 2 to the power k minus 1 that is 7 so here you can see 0 1 2 3 4 5 6 this is the window at the sender's end on the other hand at the receiver send the window is only 1 so it is now ready to receive the frame 0 and the sender sends the frame 0. As it sends the frame 0 the window is shrinking by 1. That means now it is it has got six more frames to send starting with 1 2 3 4 5 6 so when a frame is sent the window shrinks on the other hand when an acknowledgement is received the window is expanded on this side.

After the receiving the frame the receiver's side moves the window from 0 to 1 because now the frame 0 has been received and now it is ready to receive the frame number 1. So as the frame number 1 is sent again the window shrinks by 1 so it is 2 3 4 5 6 so these are the frames which it can send now.

Suppose the receiver's side sends acknowledgement ACK2 that means it is now ready to receive the frame 2. When this is received the window has been extended by 2. Therefore earlier it was ending with six now it is seven and so again the window is now having seven frames seven numbers 2 3 4 5 6 7 and 0 and now it will send frame number 2 from this end. So frame number two is sent again the window is being extended to 3 so in this way it goes on.

A piggybacking technique is used whenever two sides are communicating with each other.

(Refer Slide Time: 23:12)



Indian Institute of Technology, Kharagpur

Piggybacking

2^k-1

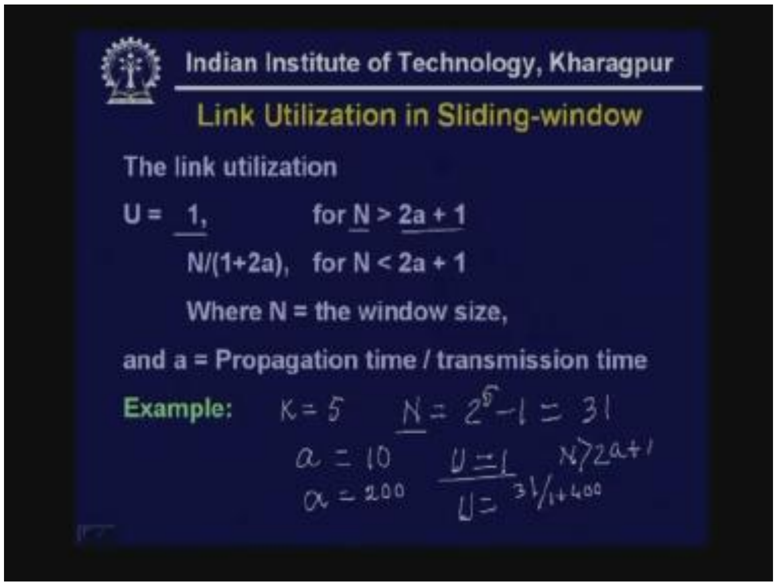
- The actual window size need not be the maximum possible size for a given sequence number length.
  - For a 3-bit sequence number, a window size of 4 can be configured.
- If two stations exchange data, each need to maintain two windows. To save communication capacity, a technique called piggybacking is used.
  - Each data frame includes a field that holds the sequence number of that frame plus a field that holds the sequence number used for ACK.
  - If a station has an ACK but no data to send, it sends a separate ACK frame.

Another point is the maximum window size need not be equal to 2 to the power of k minus 1 this is the limit for maximum size that can be used. However, in practice one can use lesser than that but in that case the overhead increases. Unnecessarily if you use more bits to represent the sequence number the overhead increases so usually we try to use the maximum number that is 2 to the power k minus 1 that is the window size.

Now, if two stations exchange data each need to maintain two windows to save communication capacity known as piggybacking. How it works? Each data frame includes a field that holds the sequence number of that frame plus a field that holds the sequence number used for acknowledgement. That means each side will have two fields. One field is the sequence number that is the frame number and another is acknowledgement number. And now the acknowledgement number is piggybacked as part of the frame that is being sent and as a result we need not send a separate frame acknowledgement for this purpose. Therefore saving in the bandwidth occurs in this way known as piggybacking.

However, if a station has an acknowledgement but no data to send it sends a separate acknowledgement frame. It is not always necessary that both sides will send acknowledgement frames by using piggybacking because if a particular station has no frame to send then it has to send acknowledgement frame after receiving the information frames. Let us see the link utilization in case of sliding-window protocol.

(Refer Slide Time: 25:30)



Indian Institute of Technology, Kharagpur

Link Utilization in Sliding-window

The link utilization

$U = \underline{1}$, for $\underline{N} > \underline{2a + 1}$

$N/(1+2a)$, for $N < 2a + 1$

Where N = the window size,

and a = Propagation time / transmission time

Example: $K = 5$ $\underline{N} = 2^5 - 1 = 31$

$a = 10$ $\underline{U = 1}$ $N > 2a+1$

$a = 200$ $U = 31/1+400$

Here the utilization will be 1 if N is greater than 2a plus 1 where N is the window size and ‘a’ is ratio between the propagation time and transmission time. Let us take an example. Suppose we are taking k is equal to 5 bit such that the value of N will be equal to 2 to the power 5 minus 1 is equal to 31. This is the window size (Refer Slide Time: 26:00) now suppose the value of a is equal to 10 let us assume so whenever a is equal to 10 obviously the utilization will be in this case 1 because N is greater than 2a plus 1, ‘a’ is equal to 10 so it is 21 so this is greater than 1 so in this case the utilization will be 1.

On the other hand let us assume the propagation time is very long and as a consequence let us assume a is equal to 200 that can happen in satellite communication. So whenever that happens the utilization will become equal to…… that is your N is equal to 31 and 1/1 plus 400 so you see the utilization is much less in such a situation whenever your are using communication over a longer distance that means propagation time is long or you are sending at a very high speed then also the transmission time will be short.

Therefore whenever N is better than 2a plus 1 in such a situation sliding-window protocol is capable of sending continuously without almost without waiting because we have seen that it maintains a window and before the window is exhausted all the frames of the window are transmitted and it receives an acknowledgement so it keeps on expanding the window in one direction. So in this way continuous flow of traffic possible if the value of N is chosen in such a way that it is greater than 2a plus 1. so you see that link utilization can be significantly better in case of sliding-window protocol.

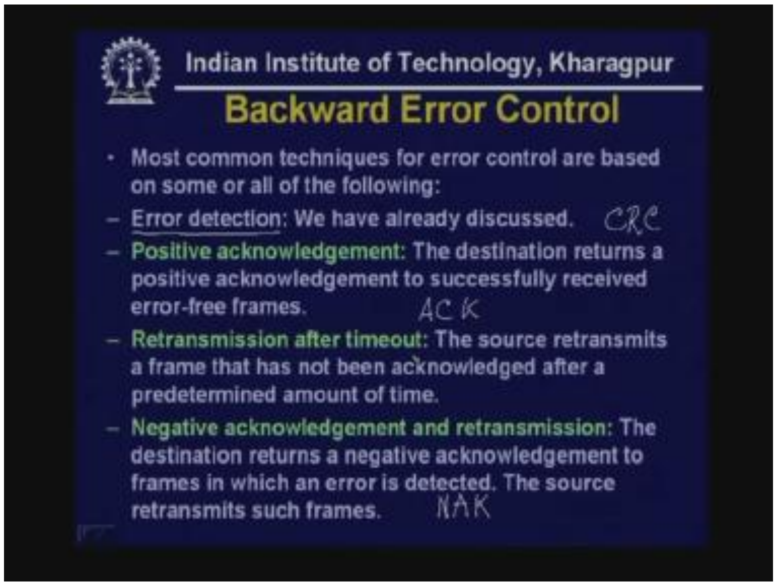
Now let us focus on error control. We shall consider this model for Backward Error Control. Data is sent as a sequence of frames as it is already been done. Frames arrive at the same order as they are sent. We are assuming that frames are reaching the destination in the same order because usually we are considering the case where the two nodes A and

B are linked by a direct communication link so in such a situation there is no possibility that the frames will be reaching out of order. It will be reaching in the same order. However, each transmission frame suffers arbitrary and variable amount of delay before reception. There can be some delay on the path because of some reason. In addition to the above following two types of errors may occur. Now what can happen is so far we have assumed that when a frame is sent it is reaching the destination without any error or without any loss however it can happen.

First of all let us consider the case of lost frame. A frame fails to arrive at the other side. How it can happen? Whenever a frame is sent in what situation a frame cannot reach the destination or it gets lost on the way? One possibility is that there may be some noise and that noise can be such that it may corrupt to such an extent that the information may become unrecognizable. So a noise burst may damage a frame to such an extent that it is not recognizable at the receiving end. In such a case we may say that the frame is lost.

Another possibility is that the frame is damaged. Whenever error occurs that is that is detected by suitable error detection scheme and then the frame is recognizable but some bit errors occur. So in such a case we call it damaged frame. So we have to take into consideration these two situations whenever we consider the error control. Most of the common techniques for error control are based on some or all of the following.

(Refer Slide Time: 31:57)



Indian Institute of Technology, Kharagpur

Backward Error Control

- Most common techniques for error control are based on some or all of the following:
  - Error detection: We have already discussed. CRC
  - Positive acknowledgement: The destination returns a positive acknowledgement to successfully received error-free frames. ACK
  - Retransmission after timeout: The source retransmits a frame that has not been acknowledged after a predetermined amount of time.
  - Negative acknowledgement and retransmission: The destination returns a negative acknowledgement to frames in which an error is detected. The source retransmits such frames. NAK

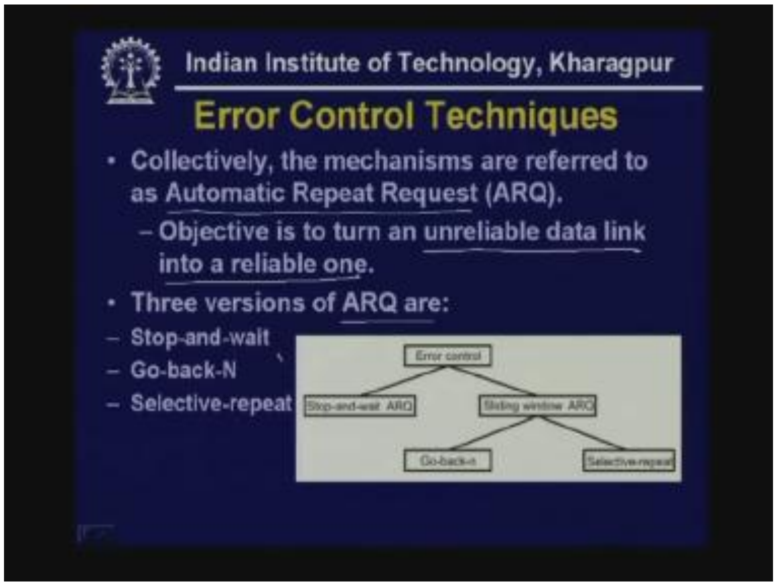
First of all it uses error detection.
What are the components of error control?
First of all there should be some technique for error detection. We have already discussed about a number of techniques for error detection such as parity check, cyclic redundancy check and checksum so one of the techniques can used for error detection. Normally the cyclic redundancy check is used because we have seen that this is the most sophisticated

one and gives you much higher fault coverage. Then it will use the concept of positive acknowledgement. The destination returns a positive acknowledgement to successfully received error free frames. That is usually represented by ACK. On the other hand there can be negative acknowledgement so the destination returns a negative acknowledgement to frames in which an error is detected but the source retransmits such frames.

So whenever an acknowledgement comes in the form of negative acknowledgement then the sender has to retransmit the frame. Apart from that there will be necessity for retransmission after time-out. The source retransmits a frame that has not been acknowledged after a predetermined amount of time. That happens in case of damaged frames. A frame is damaged to a certain extent that it is not recognizable. So the receiver assumes that no frame has come. So in such a case the sender will use some kind of time-out mechanism for retransmission. These are basic techniques or components are used for Backward Error Control.

These error control techniques are collectively referred to as Automatic Repeat Request ARQ. Obviously as I mentioned in the beginning our objective is to turn unreliable data link into a reliable one.

(Refer Slide Time: 33:20)



Indian Institute of Technology, Kharagpur

Error Control Techniques

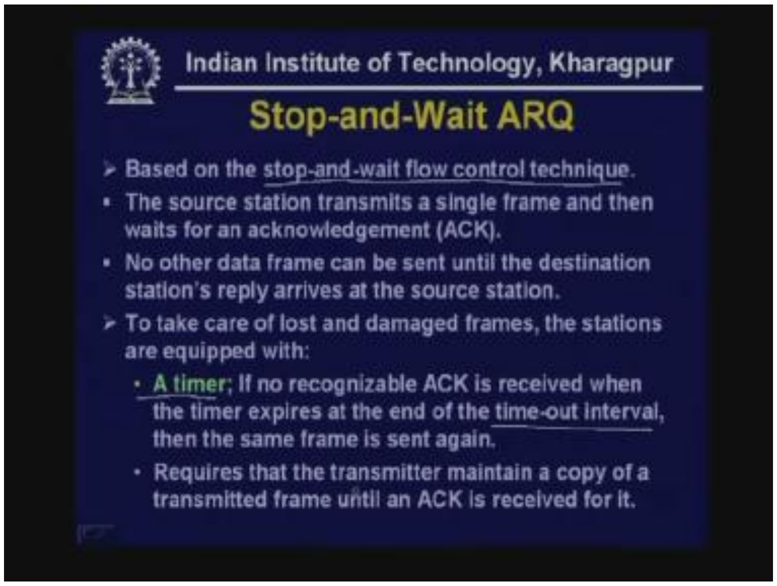
- Collectively, the mechanisms are referred to as Automatic Repeat Request (ARQ).
  - Objective is to turn an unreliable data link into a reliable one.
- Three versions of ARQ are:
- Stop-and-wait
- Go-back-N
- Selective-repeat

Error control
Stop-and-wait ARQ
Sliding window ARQ
Go-back-n
Selective-repeat

As we know there does not exist any transmission media where error cannot occur. Error can always occur although the reliability of the transmission media is improving with time. However, there is always some possibility or probability of error. So in such a case in the error prone environment our objective is to perform reliable communication and that can be done by using error control techniques. There are three versions of ARQ techniques; one is known as stop-and-wait based on stop-and-wait flow control, second is go-back-N based on sliding-window protocol then selective-repeat which is also based on sliding-window protocol. So these are the three ARQ techniques which are used and we shall discuss them one after the other.

Let us consider the stop-and-wait ARQ technique. As I mentioned it is based on stop-and-wait flow control technique. The source station transmits a single frame and then waits for an acknowledgement. As we have seen that is done in case of flow control. Then no other data can be sent until the destination stations reply arrives at the source station. This is being performed as part of the flow control. Now, to take care of lost and damaged frames the stations are equipped with another device known as timer.

(Refer Slide Time: 35:04)



Indian Institute of Technology, Kharagpur

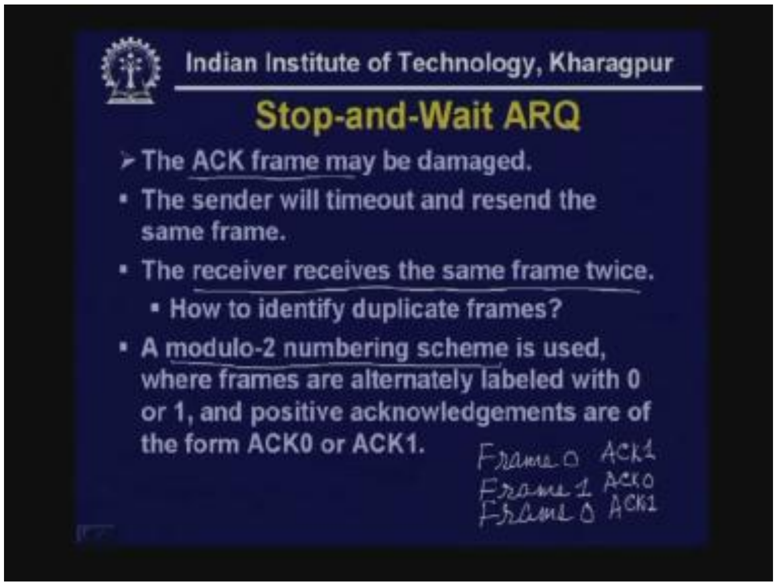
Stop-and-Wait ARQ

- Based on the stop-and-wait flow control technique.
  - The source station transmits a single frame and then waits for an acknowledgement (ACK).
  - No other data frame can be sent until the destination station's reply arrives at the source station.
- To take care of lost and damaged frames, the stations are equipped with:
  - A timer; If no recognizable ACK is received when the timer expires at the end of the time-out interval, then the same frame is sent again.
  - Requires that the transmitter maintain a copy of a transmitted frame until an ACK is received for it.

If no recognizable acknowledgement is received when the timer expires at the end of the time-out interval then the same frame is sent again. That means after sending each frame the sender starts a time-out and based on the propagation time the distance that time-out timer is fixed and after the timer reaches the time-out and if no recognizable acknowledgement is received then the sender will retransmit the frame that is the basic mechanism. So this however requires that the transmitter maintains a copy of transmitted frame until an acknowledgement is received for it.

So we have seen that sliding-window is moving forward only after receiving an acknowledgement. That means for all the frames it receives an acknowledgement the buffer can be released that is the basic idea here. However, there is a possibility that acknowledgement frame is also damaged. So in this case also the sender will time-out and resend the same frame. now there is a possibility that whenever the acknowledgement is damaged or lost then the sender will retransmit it so there is a possibility that receiver receives the same frame twice. How do your over come that? That can be overcome by using a modulo-2 numbering scheme. What is done is the frames are sent alternately as frame 0, than frame 1 then again frame 0 so alternately this is done. So if acknowledgement of frame 0 is lost then frame 0 will be transmitted so there is no possibility of confusing with the next frame because next frame will be frame 1.

(Refer Slide Time: 36:55)



Indian Institute of Technology, Kharagpur

Stop-and-Wait ARQ

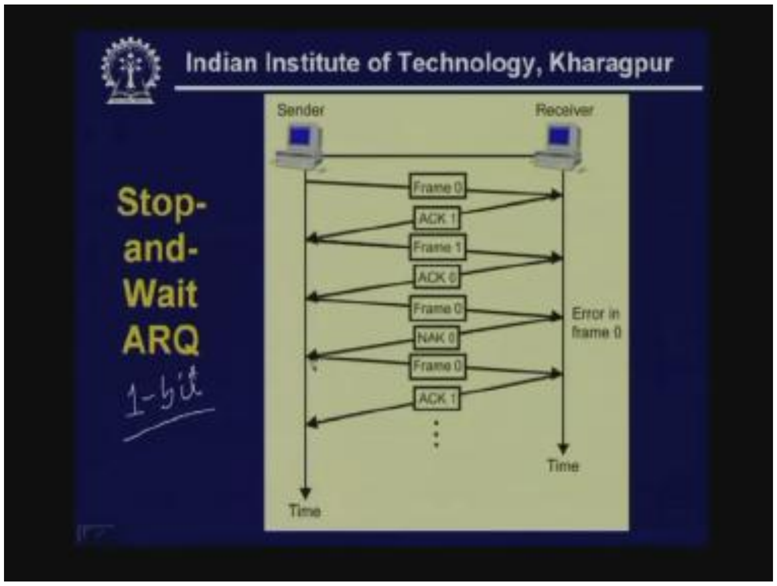
- The ACK frame may be damaged.
  - The sender will timeout and resend the same frame.
  - The receiver receives the same frame twice.
    - How to identify duplicate frames?
  - A modulo-2 numbering scheme is used, where frames are alternately labeled with 0 or 1, and positive acknowledgements are of the form ACK0 or ACK1.

Frame 0	ACK1
Frame 1	ACK0
Frame 0	ACK1

So by using this numbering the possibility of receiving duplicate frame is avoided. Similarly the acknowledgements are also numbered in this way. For example, an acknowledgement of frame 0 will be ACK1 and as an acknowledge of frame 1 it will be ACK0 and then again acknowledgement of next frame 0 will be ACK1 so this is how it goes on.

This is shown with the help of this diagram.

(Refer Slide Time: 38:10)

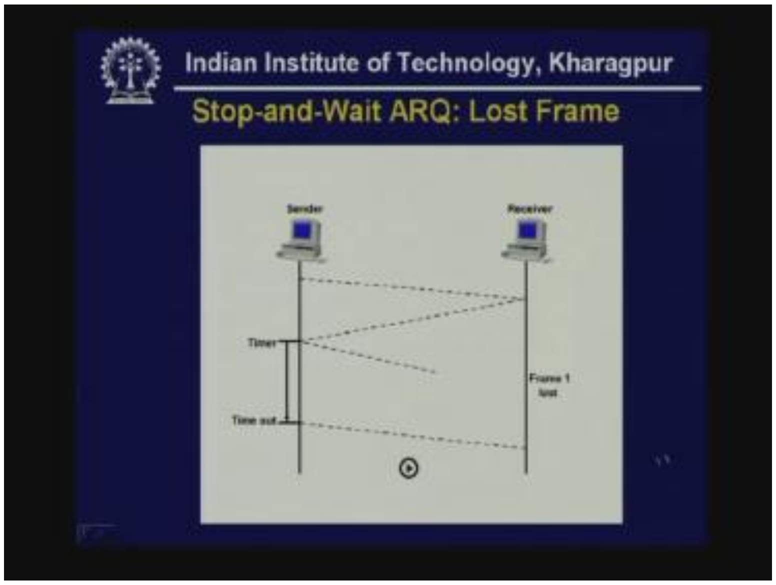


As you can see here the frame is same frame 0 and in response to that the receiver will send an acknowledgement ACK1 then the transmitter or the sender will send frame 1 and in response to that receiver will send an acknowledgement ACK0 and after receiving that acknowledgement it will send another frame that is frame 0 but these two frames are not same they are different frames. They numbered alternately as sequence number in this manner.

You may be asking why you were using modulo-2 numbering. The reason for this is we require only one bit so the overhead is much lesser, only one bit is required for numbering difference. Now you see this frame is reached with error. So receiver detects error in it so NAK 0 is set No Acknowledgement 0 is set. After receiving that the transmitter will send frame 0 once again so now it receives frame that frame in correct form so it sends an acknowledgment ACK1 then the sender will send the frame 1. So this goes on in this manner.

Now let us consider the situation when a frame is lost in transit. As I mentioned a frame may get damaged to such extent that it is not recognizable by the receiving end and in such a case we call it lost frame.

(Refer Slide Time: 39:20)

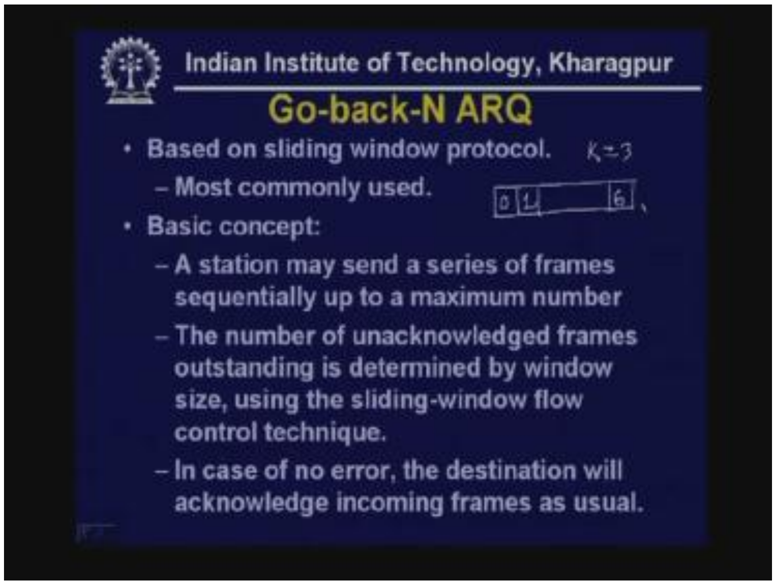

Indian Institute of Technology, Kharagpur
Stop-and-Wait ARQ: Lost Frame
Sender
Receiver
Timer
Time out
Frame 1 lost

We see that frame 0 is being sent to the receiver and an acknowledgment is sent and another frame is sent however this frame is damaged. And as a result this frame is lost and we can see there is a timer which times-out and after the time-out is reached the same frame is retransmitted so frame 0 is again retransmitted whenever the frame is lost after the time-out is reached. I have seen only in this case but it happens for all the cases.

Now let us consider the case of lost acknowledgement. in the previous case we have seen the lost frame but here we see the lost acknowledgement. So the frame is reaching without problem but the acknowledgement is now lost, timer is proceeding and as the time-out is reached again the frame 0 is sent. This is how retransmission takes place in case of stop-and-wait ARQ technique.

The main advantage of stop-and-wait ARQ technique is that it is very simple and also it requires minimum buffer size of 1 and also the number of bits required is 1. That means the number required for representing the sequence number is 1. So in both cases it is 1 so it will be minimum. However, the disadvantage is that it is highly in efficient use of the communication links particularly when 'a' is large because efficiency will be same as that stop-and-wait flow control. So we find that stop-and-wait ARQ technique is not very efficient. How it can be overcome? Obviously it can be overcome by using the technique of sliding-window protocol for ARQ. Here (Refer Slide Time: 41:00) this is known as go-back-N ARQ. Why it is called go-back-N ARQ will be clear in very soon. This is the most commonly used technique.

(Refer Slide Time: 40:55)



Indian Institute of Technology, Kharagpur

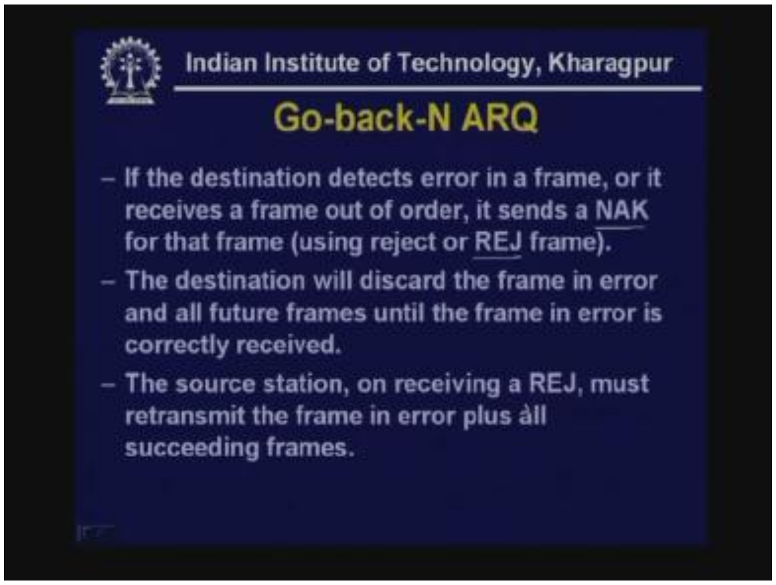
Go-back-N ARQ

- Based on sliding window protocol. $k=3$
  - Most commonly used.
- Basic concept:
  - A station may send a series of frames sequentially up to a maximum number
  - The number of unacknowledged frames outstanding is determined by window size, using the sliding-window flow control technique.
  - In case of no error, the destination will acknowledge incoming frames as usual.

The basic concept is a station may send a series of frames sequentially up to a maximum number. The maximum is equal to 3 equal to three in such a case the window size is 7 so the numbers are 0, 1 and up to 6. Therefore so many frames can be sent with receiving an acknowledgement. So 0, 1, 2 all these frames can be sent without receiving any acknowledgement basically that is the idea. The number of unacknowledged frames outstanding is determined by window size using the sliding-window flow control technique as we have already explained.

In case of no error the destination will acknowledge in the coming frames as usual. So the acknowledgements are same as usual as in the previous case. However, whenever the destination detects an error in the frame that negative acknowledgement is sent which is known as reject frame REJ frame. The destination will discard the frame in error and all future frames until the frame in error is correctly received.

(Refer Slide Time: 42:33)

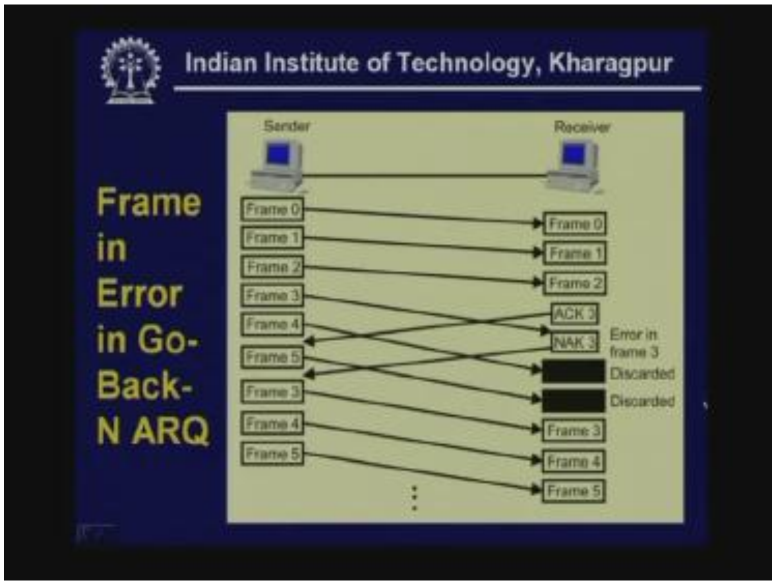

Indian Institute of Technology, Kharagpur

Go-back-N ARQ

- If the destination detects error in a frame, or it receives a frame out of order, it sends a NAK for that frame (using reject or REJ frame).
- The destination will discard the frame in error and all future frames until the frame in error is correctly received.
- The source station, on receiving a REJ, must retransmit the frame in error plus all succeeding frames.

What will happen is the destination will discard the already received frame until a particular frame is received correctly. The source station on receiving REJ reject frame must retransmit the frames in error plus all succeeding frames. Let us see how it actually works. So here it is explain with the help of the diagram.

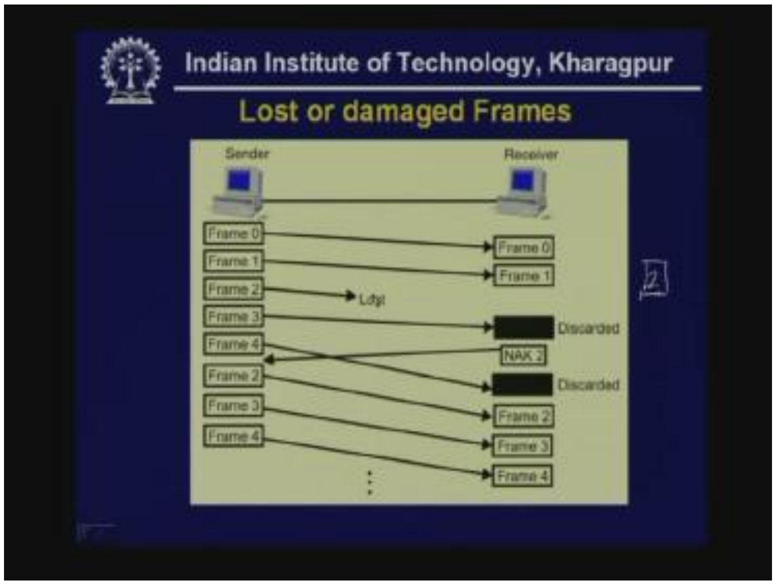
(Refer Slide Time: 43:56)


Indian Institute of Technology, Kharagpur

Frame in Error in Go-Back-N ARQ

Frame 0 is sent so here it reaches correctly, frame 1 is sent then frame 2 is sent by the sender and as you see for all these three frames a single acknowledgement is sent as ACK3. Now the transmitter is keeping on sending 3 then it is sending 4 then it is sending 5 but unfortunately the frame 3 has reached with error and as a result the receiver sends a negative acknowledgement NAK 3. At same time what it does is it discards the frame 4 and frame 5 because it has got only one buffer so there is no possibility. Since the receiver is having only one buffer it cannot really store the subsequent frames. if a frame is not received correctly then all the subsequent frames are discarded so frame 4 and 5 will be discarded so now you see that the transmitter will start sending frame 3, 4 and 5 and that is why it is called go-back-N ARQ. So it is going back to 2, 3 and again retransmitting 3, 4, 5 so 3, 4, 5 are sent and they are received one after the other at the receiving end. So the name go-back-N is coming from this. We see here that although the transmitter has sent up to 5 it is going back to 3 and again then it will be retransmitting 3, 4 and 5 so in this way it proceeds.

(Refer Slide Time: 44:38)

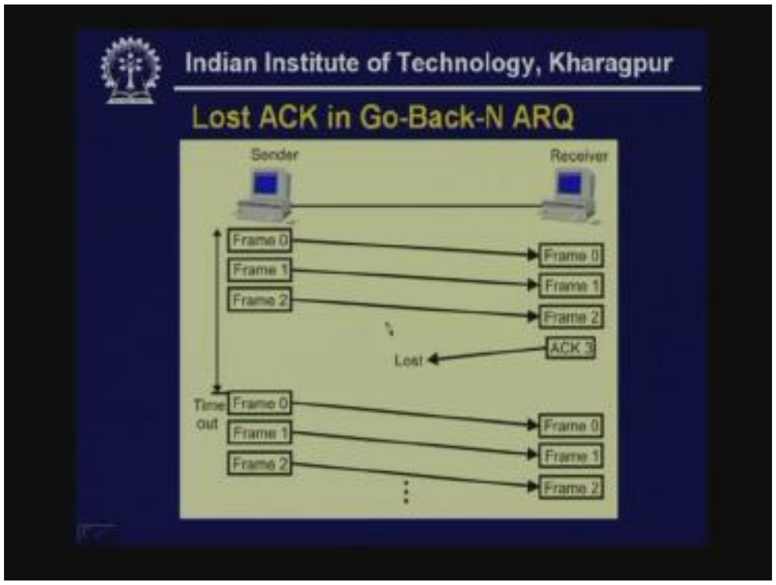

Indian Institute of Technology, Kharagpur

Lost or damaged Frames

Now let us consider what happens in case of lost or damaged frames. Here frame 0 is being sent (Refer Slide Time: 44:40) then frame 1 is being sent and frame 2 is now lost. Since frame 2 is lost and after the frame 3 is received by the destination it will be discarded because it is waiting for frame 2. The window that it was having was showing 2 so it can receive only frame 2 and no other frame can be received so here frame 3 is discarded and at the same time it sends one negative acknowledgement NAK 2. And after receiving that NAK 2 the sender will again send 2, 3 and 4 because it is using go-back-N ARQ technique. So how the lost frames are tackled is explained here. Let us see how the lost acknowledgement is tackled.

Here we see that frame 0 is sent (Refer Slide Time: 45:36) frame 1 is sent it is received correctly, frame 2 is also received correctly so it sends an acknowledgement ACK 3 that means it is asking for frame 3 to be sent and it is lost unfortunately. So after sending frame 0 it has not received acknowledgement so it has already started a timer and it reads the time-out at this point. Since no acknowledgement is received within this period retransmission starts with frame 0.

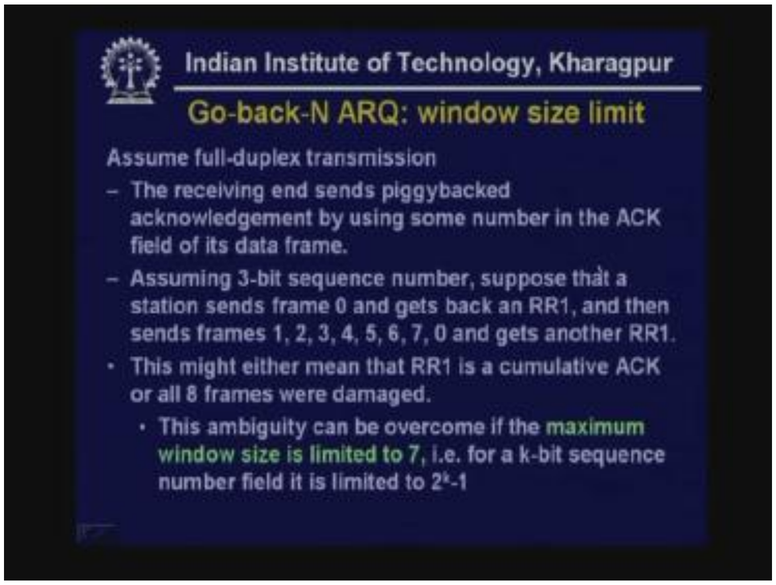
(Refer Slide Time: 45:30)


Indian Institute of Technology, Kharagpur

Lost ACK in Go-Back-N ARQ

So frame 0 frame 1 and frame 2 these are all sent one after the other to the receiving side. This how in go-back-N ARQ protocol a lost acknowledgement is taken care of.

(Refer Slide Time: 46:30)


Indian Institute of Technology, Kharagpur

Go-back-N ARQ: window size limit

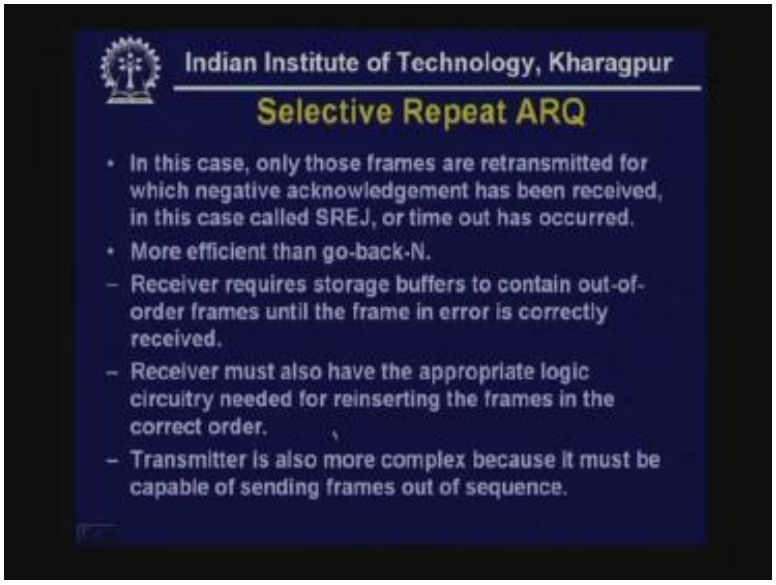
Assume full-duplex transmission
- The receiving end sends piggybacked acknowledgement by using some number in the ACK field of its data frame.
- Assuming 3-bit sequence number, suppose that a station sends frame 0 and gets back an RR1, and then sends frames 1, 2, 3, 4, 5, 6, 7, 0 and gets another RR1.
  - This might either mean that RR1 is a cumulative ACK or all 8 frames were damaged.
    - This ambiguity can be overcome if the maximum window size is limited to 7, i.e. for a k-bit sequence number field it is limited to 2^k-1

Let us see the window size limit in case of go-back-N ARQ. Let us assume full-duplex transmission is going on. The receiving end sends piggybacked acknowledgement by using some number in the acknowledgement field of data frame. Let us assume that a 3-bit sequence number is used and obviously the window size can be up to 8 so it can be 0 to 7. Now suppose that a station sends frame 0 and gets back received request 1 that is, essentially this is acknowledgement 1. Then the sender sends the frames 1, 2, 3, 4, 5, 6, 7 and 0 and gets another RR1 so it has received two RR1s. Now this might either mean that RR1 is the cumulative acknowledgement or all eight frames were damaged so the transmitter will be puzzled. So if all are damaged and lost then obviously this RR1 corresponds to the first 0. That means before this frame 0 was sent and it corresponds to frame 0.

On the other hand if it is not damaged then this RR1 can correspond to the next frame after this 0. So this (Refer Slide Time: 48:05) can be either for this or the next frame after 0. So the transmitter gets confused in selecting the right frame. This ambiguity can be overcome if the maximum window size is limited to 7. That means if the window size is limited to 7 that is 0 to 6 then this problem would not arise it can be very easily proved. That is why whenever k bit sequence number is used the window size is limited to 2 to the power k minus 1 in case of go-back-N ARQ.

Let us see the case of Selective-Repeat ARQ technique. In this case only those frames are retransmitted for which negative acknowledgement has been received. This negative acknowledgement is referred to as SREJ Selective Reject or time-out has occurred. In either case only that frame is sent. We have seen that in case of go-back-N the transmitter is going back to some previous number and it sends a number of frames which were already retransmitted that is not done in case of Selective-Repeat ARQ and as a consequence it is more efficient than go-back-N ARQ. However, in this case the receiver requires storage buffers to contain out of order frames until the frame in error it correctly receives.

(Refer Slide Time: 48:42)


Indian Institute of Technology, Kharagpur

Selective Repeat ARQ

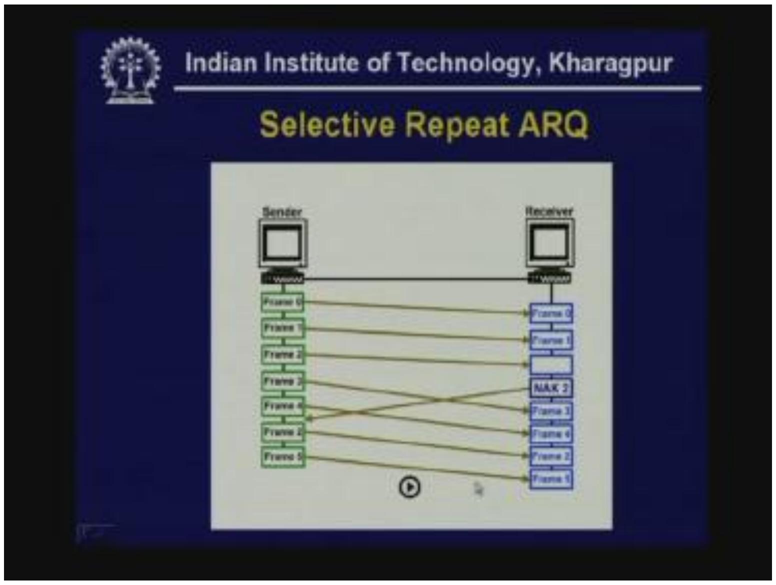
- In this case, only those frames are retransmitted for which negative acknowledgement has been received, in this case called SREJ, or time out has occurred.
- More efficient than go-back-N.
  - Receiver requires storage buffers to contain out-of-order frames until the frame in error is correctly received.
  - Receiver must also have the appropriate logic circuitry needed for reinserting the frames in the correct order.
  - Transmitter is also more complex because it must be capable of sending frames out of sequence.

In case of go-back-N we have sent the receiver requires only one buffer. However, in this case the number of buffers required will be more than 1. The sender also will require buffers more than one and receiver also will require buffer more than one. Moreover the receiver must have appropriate logic circuitry needed for reinserting the frames in the correct order. Transmitter is also more complex because it must be capable of sending frames out of sequence because in both cases out of order frames are to be handled. Let us see about that with the help of this animation.

Frame 0 is being sent by the transmitter and after receiving this frame the transmitter is sending another frame that is frame 1 and then frame 2 but however frame 2 reaches the destination with error where some bits get corrupted. Obviously the sender will send a negative acknowledgement although the transmitter will keep on sending subsequent frames. We see that negative acknowledgement is reaching the destination now. So frame 2 has to be sent and retransmitted by the sender and before that already frame 3 and frame 4 have been sent.

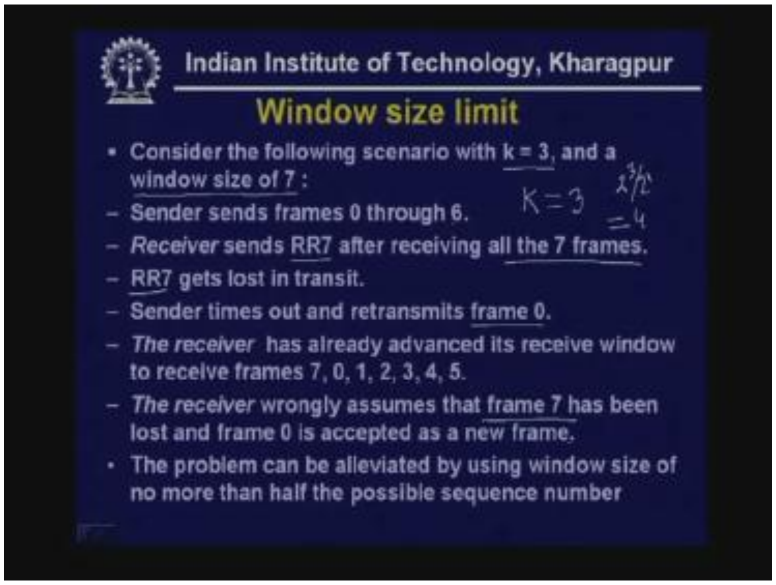
Hence in case of go-back-N ARQ after 2 the 3, 4 and 5 are sent but now here we find that after sending 2 it sends 5 so retransmission of 3 and 4 is prevented by using Selective-Repeat ARQ technique. However here the receiver must keep buffer so that 2 can be inserted before 3 and 4 and also it must have little more possessing capability.

(Refer Slide Time: 51:40)



Therefore we find that in case of selective-repeat with the help of more powerful processor capability and more number of buffers in the receiving end the efficiency can be improved. Also, it will require some limit on the window size. Let us consider the following scenario in case of Selective-Repeat ARQ assuming k is equal to 1 and let us assume the window size is same as go-back-N ARQ.

(Refer Slide Time: 52:10)



Indian Institute of Technology, Kharagpur

Window size limit

- Consider the following scenario with k = 3, and a window size of 7 :
  - Sender sends frames 0 through 6.
  - *Receiver* sends RR7 after receiving all the 7 frames.
  - RR7 gets lost in transit.
  - Sender times out and retransmits frame 0.
  - *The receiver* has already advanced its receive window to receive frames 7, 0, 1, 2, 3, 4, 5.
  - *The receiver* wrongly assumes that frame 7 has been lost and frame 0 is accepted as a new frame.
- The problem can be alleviated by using window size of no more than half the possible sequence number

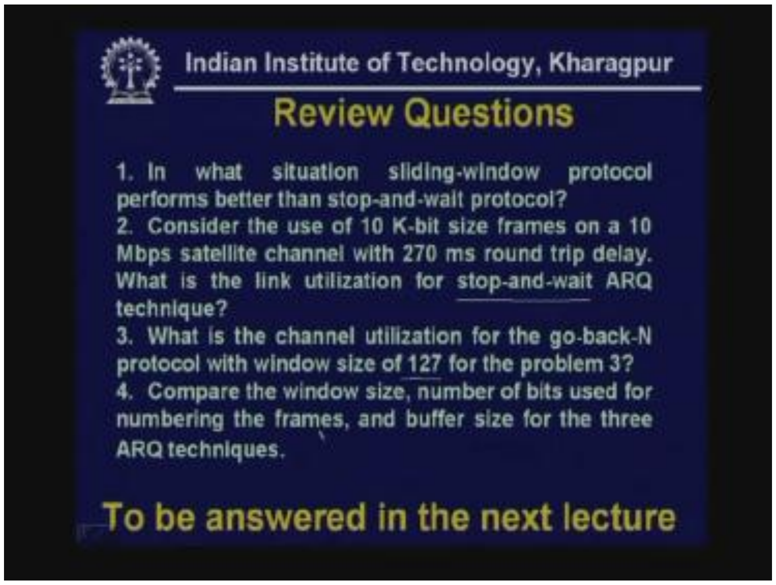
Now the sender sends frames starting from 0 to 6 one after the other. The receiver sends receive requests 7 that means after receiving 6 it will send the receive request 7 after receiving all seven frames. So unfortunately RR7 gets lost in transit though sender times-out and retransmits frame 0. Since it was waiting for acknowledgement for these frames they are not received so it will send frame 0.

However, the receiver already has received only seven frames it was also sending frame 0 but not the frame 0 of the previous one but the next one. So the receiver already advanced its received window to receive frame 7, 0, 1, 2, 3, 4, 5 so it is now ready to receive frame 0. So the receiver wrongly assumes that frame 7 has been lost and frame 0 is accepted as a new frame, instead of old frame it is received as a new frame so we see that the problem arises here. This problem can again be alleviated by choosing a limited window size and in this case the window size should be no more than half the possible sequence number. that means if you are choosing k is equal to 3 then the possible window size in case of Selective-Repeat ARQ has to be 2 to the power 3/2 that is is equal to 4.

In case of go-back-N ARQ this can be 7 however in case of Selective-Repeat ARQ it has to be 4. Moreover both in case of receiver and transmitter we require more number of bits here because it has to be 2 to the power k/2. Suppose we are having a window size of 7 so for window size 7 in case of go-back-N ARQ the number of bits required is 3. On the other hand for Selective-Repeat ARQ the number bits required will be 4. So for the same window size the Selective-Repeat ARQ technique will require more number of bits for numbering the frames.

Therefore we have discussed two flow control techniques the stop-and-wait and sliding-window technique and also we have discussed three ARQ techniques; stop-and-wait, go-back-N and selective-repeat. And as I mentioned these flow and error control are performed in an integrated manner although I have discussed them separately for the convenience of your understanding. Now it is time to give you some review questions.

(Refer Slide Time: 56:30)



Indian Institute of Technology, Kharagpur

Review Questions

1. In what situation sliding-window protocol performs better than stop-and-wait protocol?
2. Consider the use of 10 K-bit size frames on a 10 Mbps satellite channel with 270 ms round trip delay. What is the link utilization for stop-and-wait ARQ technique?
3. What is the channel utilization for the go-back-N protocol with window size of 127 for the problem 3?
4. Compare the window size, number of bits used for numbering the frames, and buffer size for the three ARQ techniques.

To be answered in the next lecture

1) In what situation sliding-window protocol performs better than stop-and-wait protocol?

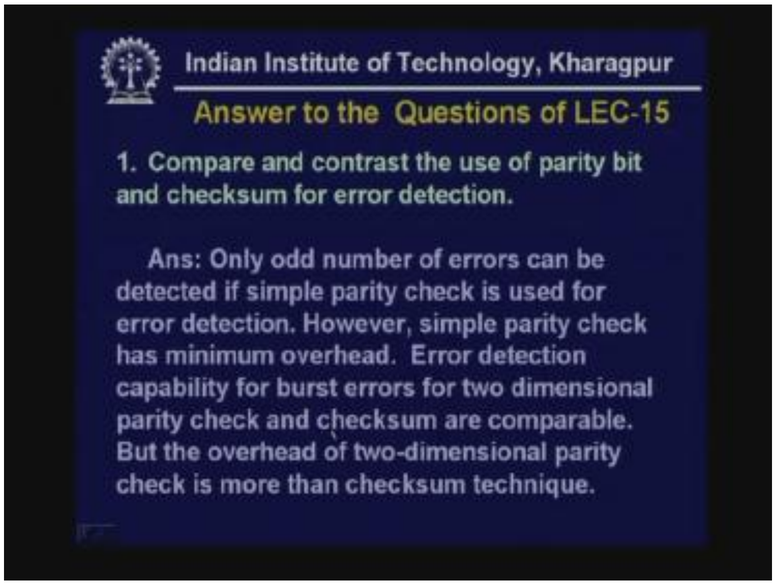
2) Consider the use of 10 k-bit size frames on a 10 Mbps satellite channel with 270 ms trip delay round trip delay. What is the link utilization for stop-and-wait ARQ technique?

3) What is the channel utilization for the go-back-N protocol with window size of 127 for the problem 3?

So in the previous case the value of n was different actually it was stop-and-wait so window size was 1 but now it is 127 so in this case what is the utilization?

4) Compare the window size number of bits used for numbering the frames and buffer size for the three ARQ techniques.

(Refer Slide Time: 57:10)



Indian Institute of Technology, Kharagpur

Answer to the Questions of LEC-15

1. Compare and contrast the use of parity bit and checksum for error detection.

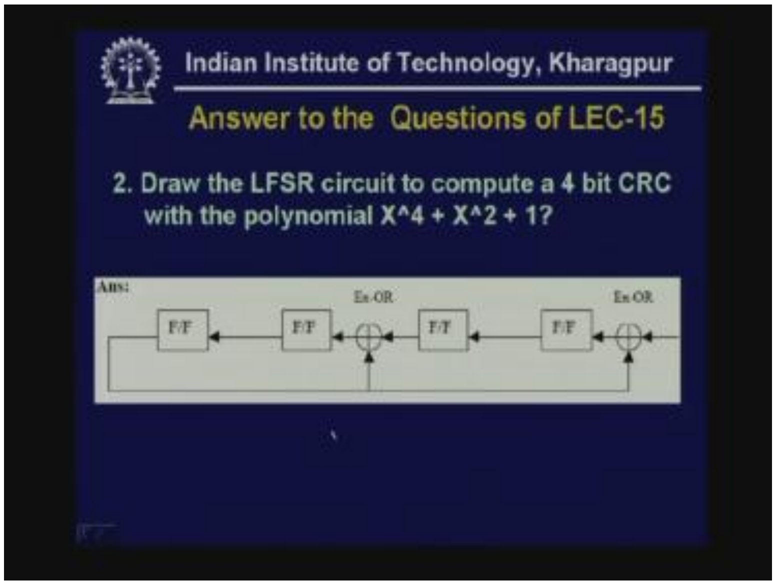
Ans: Only odd number of errors can be detected if simple parity check is used for error detection. However, simple parity check has minimum overhead. Error detection capability for burst errors for two dimensional parity check and checksum are comparable. But the overhead of two-dimensional parity check is more than checksum technique.

There are answers to questions for lecture-15.

1) Compare and contrast the use of parity bit and checksum for error detection.

Only odd number of error can be detected if simple parity check is used for error detection. However, simple parity check has minimum overhead. Error detection capability for burst errors for two dimensional parity check and checksum are comparable but the overhead of two dimensional parity check is more than checksum techniques as we have discussed in detail.

(Refer Slide Time: 57:23)

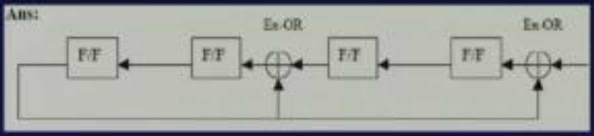


Indian Institute of Technology, Kharagpur

Answer to the Questions of LEC-15

2. Draw the LFSR circuit to compute a 4 bit CRC with the polynomial X^4 + X^2 + 1?

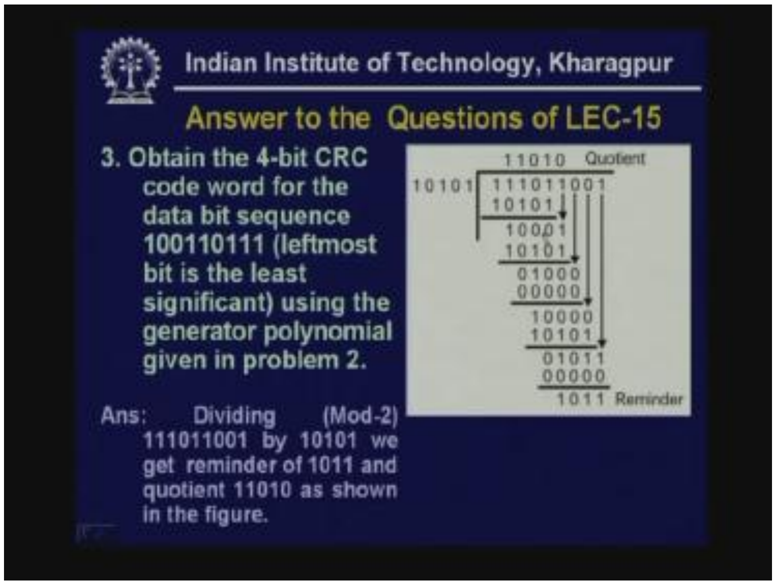
Ans:



2) Draw the LFSR circuit to compute a 4-bit CRC with the polynomial x to the power 4 plus x to the power 2 plus 1.

As you can see since the degree of polynomial is 4 we require four flip-flops and here we have got three terms so we require two Exclusive OR gates connected in this manner as it is shown in this diagram. So this is the LFSR circuit for this characteristic polynomial.

(Refer Slide Time: 57:55)



Indian Institute of Technology, Kharagpur

Answer to the Questions of LEC-15

3. Obtain the 4-bit CRC code word for the data bit sequence 100110111 (leftmost bit is the least significant) using the generator polynomial given in problem 2.

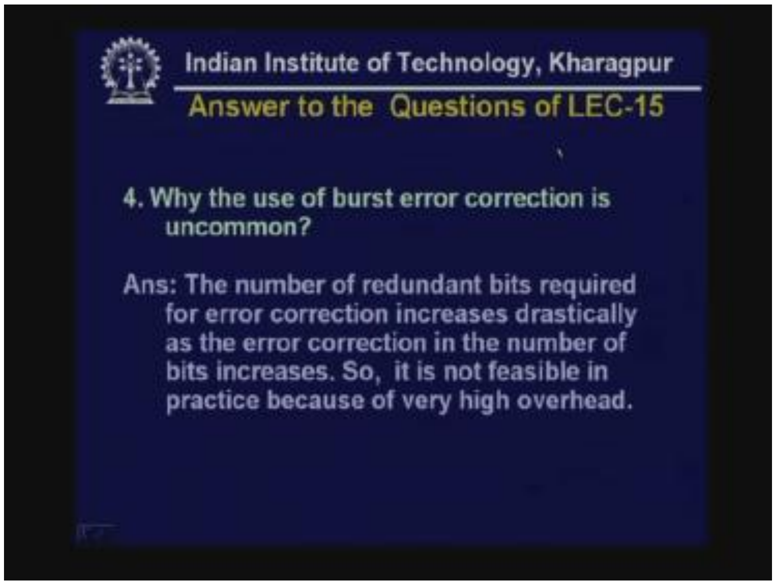
```
              11010     Quotient
       ______________
10101 | 111011001
        10101
        -----
         10001
         10101
         -----
          01000
          00000
          -----
           10000
           10101
           -----
            01011
            00000
            -----
             1011  Reminder
```

Ans: Dividing (Mod-2) 111011001 by 10101 we get reminder of 1011 and quotient 11010 as shown in the figure.

3) Obtain the 4-bit CRC code word for the data bit sequence 100110111 using the generator polynomial given in problem 2.

Here the sequence is given and I have already explained that how this number is divided by modula-2 arithmetic. as you can see first quotient is 1, the second quotient is also 1 then Exclusive OR performed bit by bit, third quotient is 0 so 0 0 and it becomes the same so in this way we will get 1011 as the reminder which is used as the CRC and this is the quotient.

(Refer Slide Time: 58:38)



Indian Institute of Technology, Kharagpur

Answer to the Questions of LEC-15

4. Why the use of burst error correction is uncommon?

Ans: The number of redundant bits required for error correction increases drastically as the error correction in the number of bits increases. So, it is not feasible in practice because of very high overhead.

4) Why the use of burst error correction is uncommon?

The number of redundant bits required for error correction increases drastically as the error correction in the number of bits increases. As we know in case of burst error the number of bits may get corrupted and as a result error correction is very very difficult and inefficient that is why it is not commonly used. So with this we come to the end of today’s lecture, thank you.