NEED FOR ERROR DETECTION AND CORRECTION

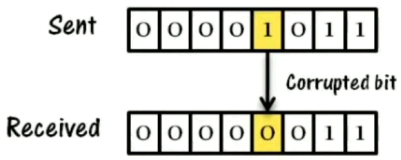
- When a data is transmitted over a network, it may get corrupted due to many factors
- One or more bits of a message can be altered
- Some mechanism is required for detecting and correcting these errors

TYPES OF ERRORS

- Single bit error
- Burst error

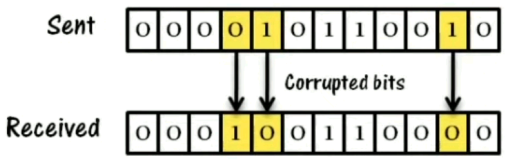
SINGLE BIT ERROR

Only 1 bit of the data unit is changed



BURST ERROR

Two or more bits of the data unit is changed

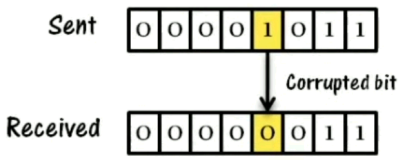


TYPES OF ERRORS

- Single bit error
- Burst error

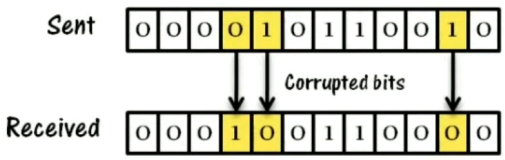
SINGLE BIT ERROR

Only 1 bit of the data unit is changed



BURST ERROR

Two or more bits of the data unit is changed



The number of bits affected depends on the data rate and duration of noise

REDUNDANCY

- To detect or correct errors, we need to send some extra bits with our data called redundant bits
- Redundant bits are added by the sender and removed by the receiver
- This concept is called Redundancy

DETECTION *VERSUS* CORRECTION

- Error detection is simple that we are looking only if the error is occurred or not
- In Error correction, we have to know exact number of bits corrupted and the location
- So, error correction is more difficult than error detection
- Two types of error correction,
  - ✓ Forward error correction
  - ✓ Backward error correction / Retransmission

CODING

- Redundancy is achieved through various coding schemes
- Two types of coding
  - ✓ Block coding
  - ✓ Convolution coding

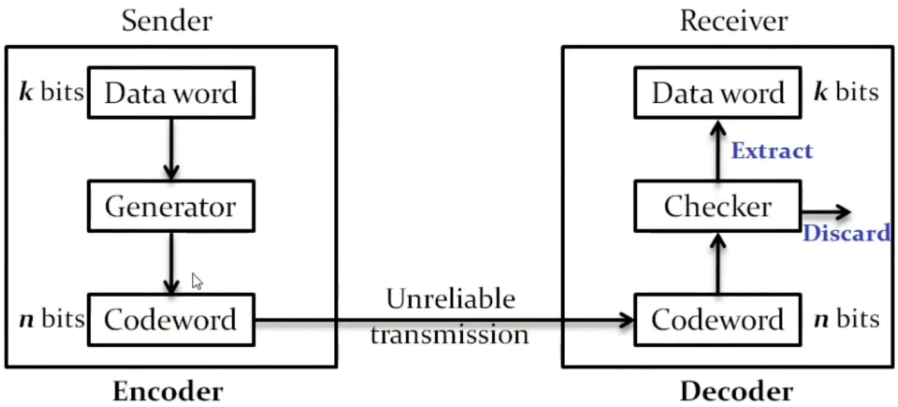
BLOCK CODING

- In block coding, we divide the message into blocks, each of k bits, called data words
- We add r redundant bits to each block to make the length

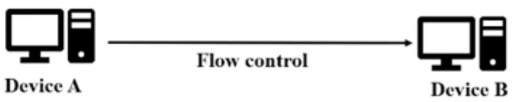
$$n = k + r$$

- The resulting n-bit blocks are called code words

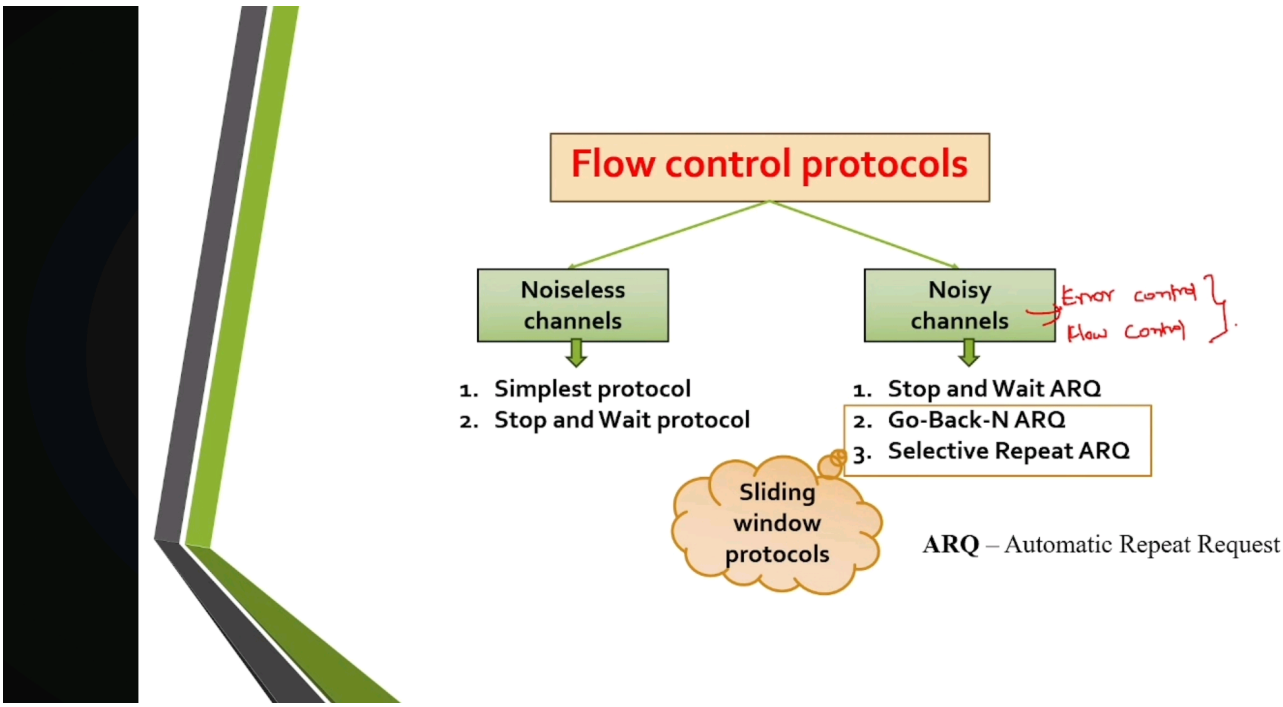
Error detection in block coding



FLOW CONTROL



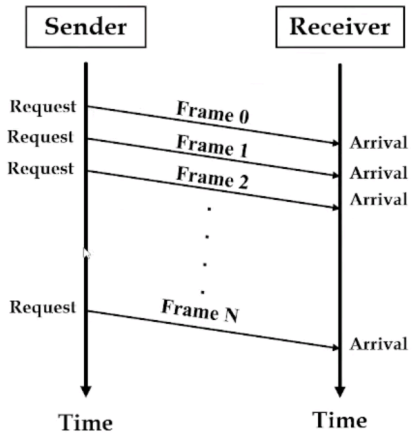
- One of the most important duties of datalink layer
- Flow control refers to set of procedures used to restrict the amount of data that a sender can send before waiting for acknowledgement from the receiver


Flow control protocols
Noiseless channels
Noisy channels
Error control
Flow control
1. Simplest protocol
2. Stop and Wait protocol
1. Stop and Wait ARQ
2. Go-Back-N ARQ
3. Selective Repeat ARQ
Sliding window protocols
ARQ – Automatic Repeat Request

SIMPLEST PROTOCOL

- No flow control and no error control
- We assume that the receiver can handle any frame it receives
- Receiver can never be overwhelmed with incoming frames
- The datalink layer of the receiver immediately removes the header from the frame and handover the date packet to the network layer that also accepts the data packet immediately
- It is a unidirectional protocol
- Data frames travel from one direction from sender to receiver

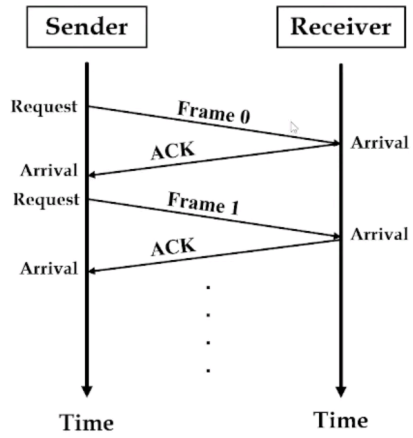
FLOW DIAGRAM



STOP AND WAIT PROTOCOL

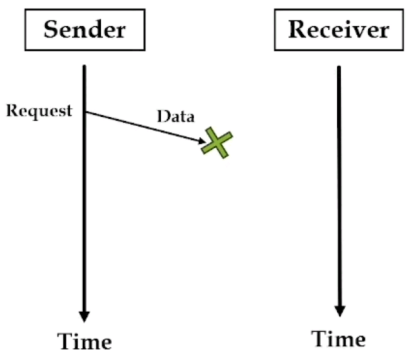
- No Error control
- Only flow control
- In stop and wait protocol, after sending one frame, the sender stops and waits for acknowledgement from the receiver before sending the next frame
- Unidirectional transmission

FLOW DIAGRAM



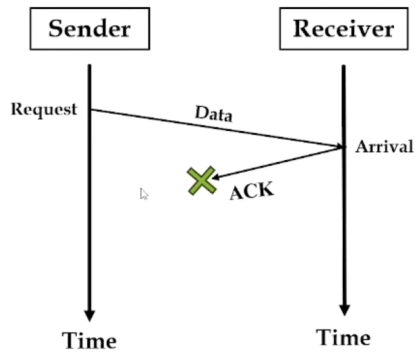
PROBLEMS IN STOP AND WAIT PROTOCOL

1. PROBLEMS DUE TO LOST DATA :



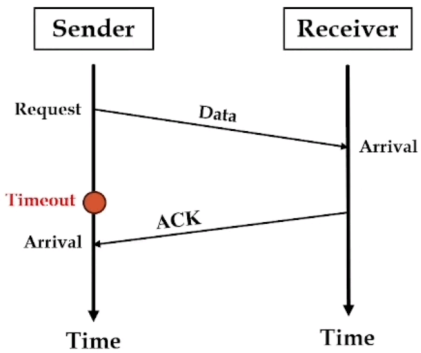
- The sender waits for the ACK for an endless time
- The receiver waits for the data frame for an endless time

2. PROBLEMS DUE TO LOST ACKNOWLEDGEMENT :



- The sender waits for the ACK for a long time

3. PROBLEMS DUE TO DELAYED DATA OR ACKNOWLEDGEMENT:



- When the ACK is received late after timeout, it can be wrongly considered to be the ACK of any other frame

STOP AND WAIT ARQ PROTOCOL

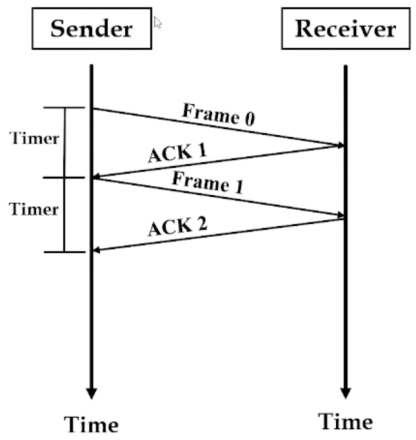
- In stop and wait ARQ protocol, the sender sends one frame at a time and waits for an acknowledgement (ACK) from the receiver before sending the next frame
- To detect errors, error control mechanism is added to each frame
- When the data frame arrives at the receiver side, it is checked for errors and if it has any errors, the frame is discarded

Stop and wait ARQ = Stop and wait + Timeout + Sequence number

- Every time when a sender sends the frame, it starts the timer
- When ACK arrives before the timer expires, the timer is stopped and the sender sends the next frame
- If the timer expires (Timeout), the sender resends the previous frame, assuming that the frame was either corrupted or lost
- To prevent duplicate frames, sequence numbers and acknowledgement numbers are added
- The acknowledgement numbers always announce the sequence number of the next frame expected by the receiver

WORKING OF STOP AND WAIT ARQ PROTOCOL :

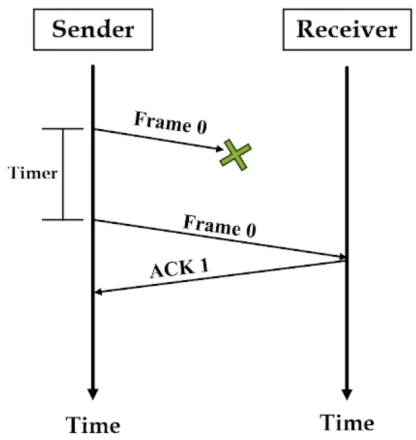
Case 1:



- The sender sends Frame 0 and waits for the ACK from receiver
- The receiver receives Frame 0 and sends ACK 1 to sender
- After receiving the ACK, the sender sends Frame 1 and this process is continued until all frames are received by the receiver

WORKING OF STOP AND WAIT ARQ PROTOCOL :

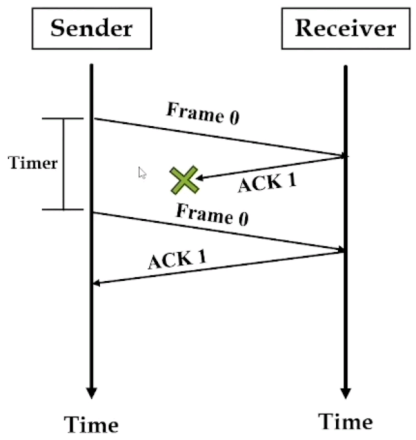
Case 2:



- Here, Frame 0 is lost, so no ACK is sent from the receiver side
- When the timer expires, the sender retransmits the Frame 0
- Here, the sender has to keep the copy of the frame until its ACK arrives

WORKING OF STOP AND WAIT ARQ PROTOCOL :

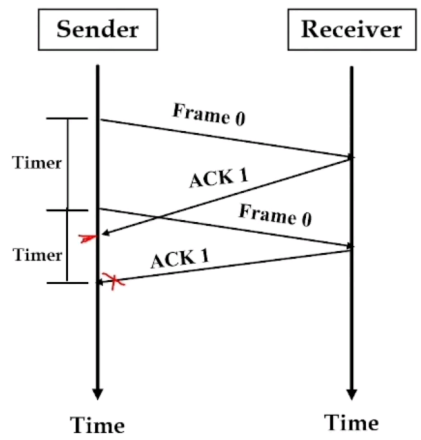
Case 3:



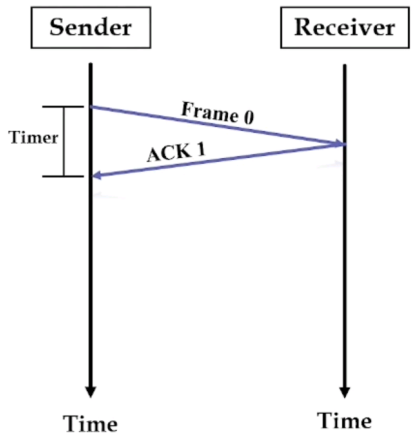
- Here, ACK 1 is lost
- When the timer expires, the sender retransmits the Frame 0
- Here, the receiver discards the duplicate frame

WORKING OF STOP AND WAIT ARQ PROTOCOL :

Case 4:



- Here, Frame 0 is received by the receiver and ACK 1 is sent
- But the ACK is received at sender side after the timer expires
- So, when timer expires sender retransmits Frame 0
- Here, the sender discards the duplicate ACK



Round Trip Time (RTT):

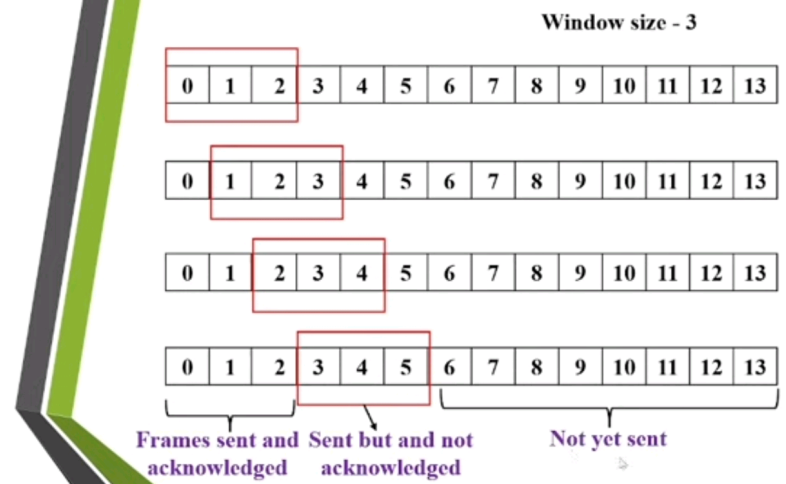
The time taken by the data frame to reach the receiver + the time taken by the ACK to reach the sender

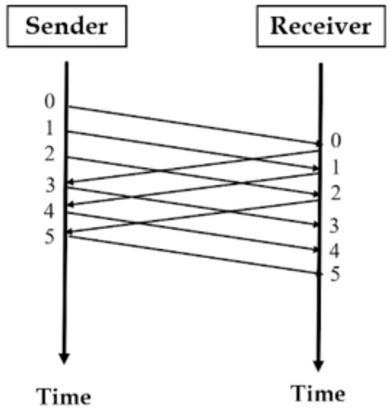
- Stop and wait ARQ protocol is a special case of Sliding window protocol
- For stop and wait ARQ protocol, the window size is 1
- In stop and wait ARQ protocol, the efficiency is very low as only one frame is sent at a time
- Hence, to improve the efficiency Go-Back-N ARQ and Selective Repeat ARQ protocols are used

SLIDING WINDOW PROTOCOL

- Multiple frames can be sent by sender before receiving the acknowledgement from receiver
- Here, the sender has a buffer called sending window and the receiver has a buffer called receiving window
- The number of frames is sent based on the window size
- Sliding window protocol is also known as windowing

WORKING OF SLIDING WINDOW PROTOCOL :





Types of sliding window protocol :

- Go-Back-N ARQ protocol
- Selective Repeat ARQ protocol

GO-BACK-N ARQ PROTOCOL

- Go-Back-N ARQ protocol is a type of sliding window protocol
- N is the sender window size
- Receiver window size is always 1
- In Go-Back-N ARQ protocol, multiple frames can be sent before receiving the acknowledgement but the receiver can buffer only one packet
- The sender has to keep the copy of sent frames until the acknowledgement arrives

- If the acknowledgement is not arrived before the timer expires, all the frames in the current window are retransmitted
- The acknowledgement number is cumulative and it defines the sequence number of the next frame expected to arrive

- The sequence numbers are in modulo 2^m
- The range of sequence number will be from 0 to 2^m-1

  Here m - size of sequence number field in bits
- The size of sending window will be 2^m-1

  ⇨ If m=2,

  No of sequence numbers = 2^2 = 4 (00,01,10,11)

  Range of sequence number = 0 to 3

  The sending window size will be 3

WORKING OF GO-BACK-N ARQ PROTOCOL :

m=3 ➡ $2^3 = 8$ ➡ 0 to 7

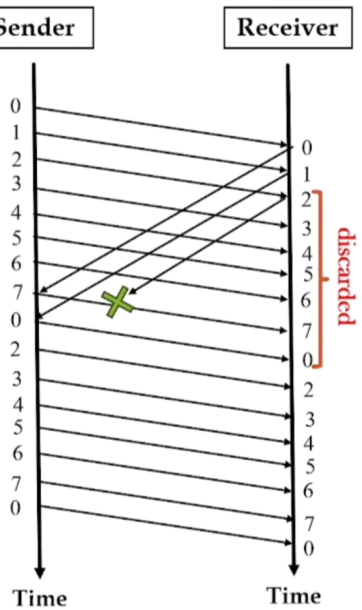
Window size = 7

0	1	2	3	4	5	6	7	0	1	2	3	4	5

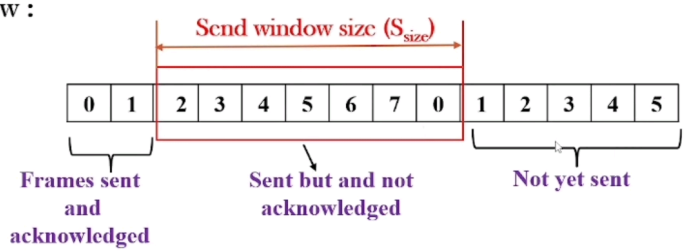
0	1	2	3	4	5	6	7	0	1	2	3	4	5

0	1	2	3	4	5	6	7	0	1	2	3	4	5

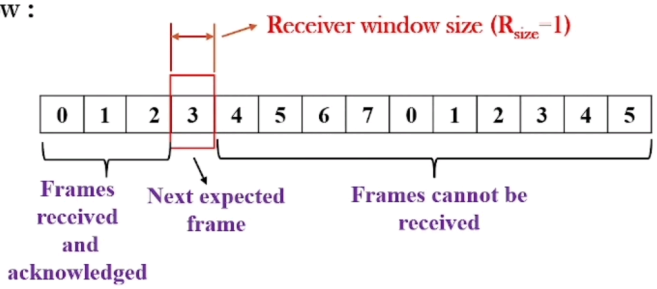
Here, the sender goes back and resends all the frames in the current window. Hence the protocol is called Go-Back-N



Send window :



Receive window :



Drawback of Go-Back-N :

- If acknowledgement of one frame is not received, the entire window is retransmitted which can cause congestion in the network and many packets can be lost

SELECTIVE-REPEAT ARQ PROTOCOL

- In Selective-Repeat ARQ protocol, only the lost or corrupted frames are retransmitted
- Selective-Repeat protocol also uses two windows : a send window and a receive window
- Here, the maximum size of send window is much smaller ;i.e., 2^{m-1}
- Receive window is same size as the send window
- In Selective-Repeat ARQ protocol the receiver sends a negative ACK for the lost / corrupted frame so that the sender retransmits the corresponding frame

- Size of send window : 2^{m-1}

➡ if m=2 , Send window = $2^{2-1} = 2^1 = 2$

Receive window =2

- The range of sequence number will be from 0 to 2^m-1

= 0 to 2^2-1 = 0 to 3

WORKING OF SELECTIVE-REPEAT ARQ PROTOCOL :

m=3 ➟ 0 to 2^3-1 ➟ 0 to 7

Window size = 2^{3-1} =4

0	1	2	3	4	5	6	7	0	1	2	3	4	5

0	1	2	3	4	5	6	7	0	1	2	3	4	5

0	1	2	3	4	5	6	7	0	1	2	3	4	5

Here, the sender resends only selective frames that are corrupted or lost. Hence the protocol is called Selective Repeat