# Part III Datalink Layer

***Chapter 10 :Error Detection and Correction***

***Chapter 11 Data Link Control and Protocols***

***Chapter 12 Multiple Access***

***Chapter 13 Wired LANs: Ethernet***

***Chapter 15 Connecting LANs, Backbone networks and virtual LAN***

# Chapter 11

# Data Link Control and Protocols

# ***Flow and Error Control***

*The most important responsibilities of the data link  layer are flow control and error control. these  functions are known as :*

*data link control.*

# *Data link control.*

- *Data must be checked and processed before they can be used.*
- *The rate of such processing is often slower than the rate of transmission.*
- *For this reason , each receiver has a buffer to store incoming data until they are processed.*
- *If buffer begin to fill up, the sender must slow or halt transmission.*

**Note**

*Flow control refers to a set of procedures used to restrict the amount of data that the sender can send before waiting for acknowledgment.*

# *Data link control.*

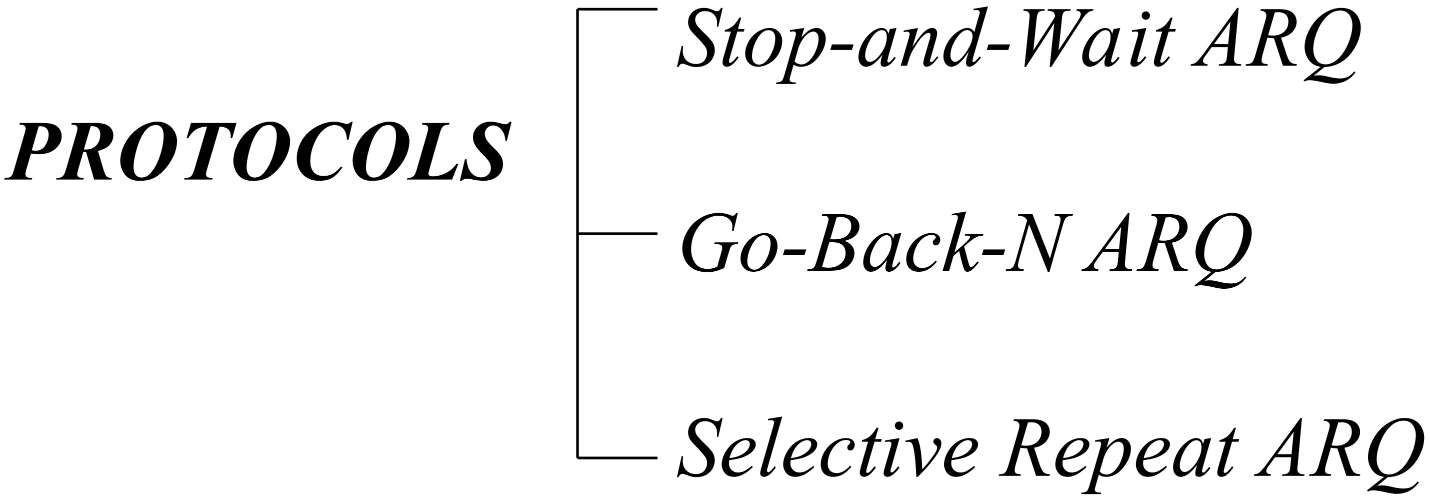
*Error control : is both error detection and correction. Error correction in data link layer is implemented simply: anytime an error is detected in exchange, specified frames are retransmitted. This process is called Automatic Repeat Request*

**Note**

*Error control in the data link layer is based on automatic repeat request, which is the retransmission of data.*

# *PROTOCOLS*

*Now let us see how the data link layer can combine flow control , and error control to achieve the delivery of data from one node to another . The protocols are normally implemented in software by using one of the common programming languages.*



**PROTOCOLS**

- *Stop-and-Wait ARQ*
- *Go-Back-N ARQ*
- *Selective Repeat ARQ*

# *Stop-and-Wait ARQ*

- *It is the simplest flow and error control mechanism . A transmitter sends a frame then stops and waits for an acknowledgment.*
- *Stop-and-Wait ARQ has the following features:*
- ✓ *The sending device keeps a copy of the sent frame transmitted until it receives an acknowledgment( ACK)*
- ✓ *The sender starts a timer when it sends a frame. If an ACK is not received within an allocated time period, the sender resends it*
- ✓ *Both frames and acknowledgment (ACK) are numbered alternately 0 and 1( two sequence number only)*
- ✓ *This numbering allows for identification of frames in case of duplicate transmission*

# *Stop-and-Wait ARQ*

- *The acknowledgment number defines the number of next expected frame. (frame 0 received ACK 1 is sent)*
- *A damage or lost frame treated by the same manner by the receiver*
- *If the receiver detects an error in the received frame, or receives a frame out of order it simply discards the frame*
- *The receiver send only positive ACK for frames received safe; it is silent about the frames damage or lost.*
- *The sender has a control variable Sthat holds the number of most recently sent frame (0 or 1). The receiver has control variable R, that holds the number of the next frame expected (0,or 1)*

# *Stop-and-Wait ARQ*

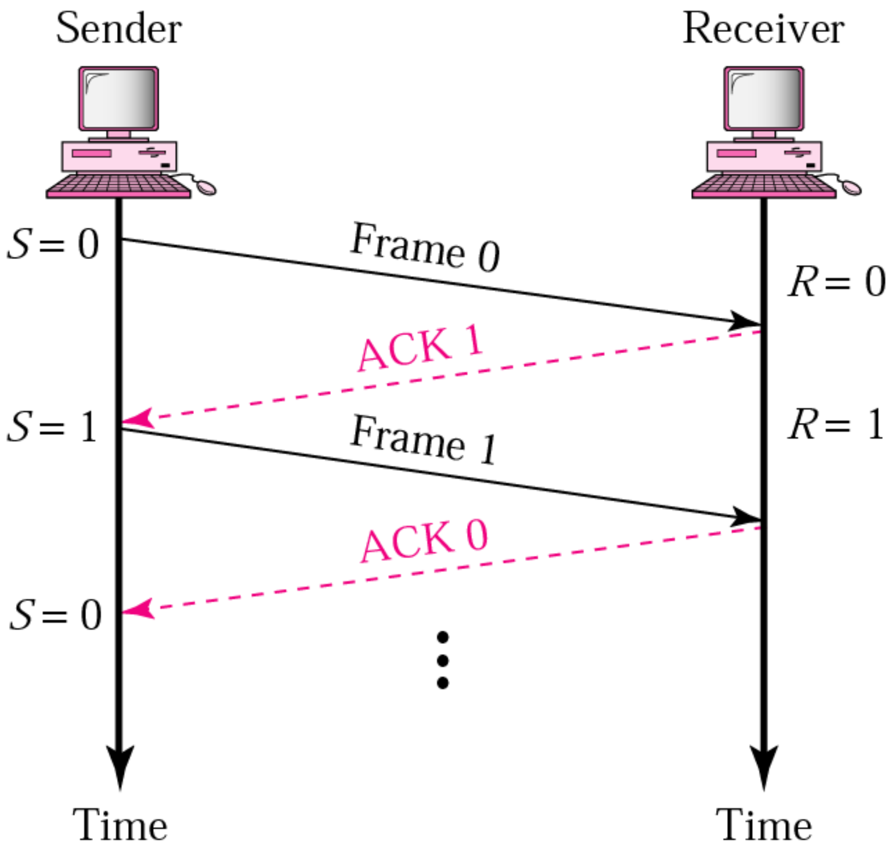
## *Cases of Operations:*

*1.Normal operation*
*2.The frame is lost*
*3.The Acknowledgment (ACK) is lost*
*4.The Ack is delayed*

# *Stop-and-Wait ARQ*

Normal operation

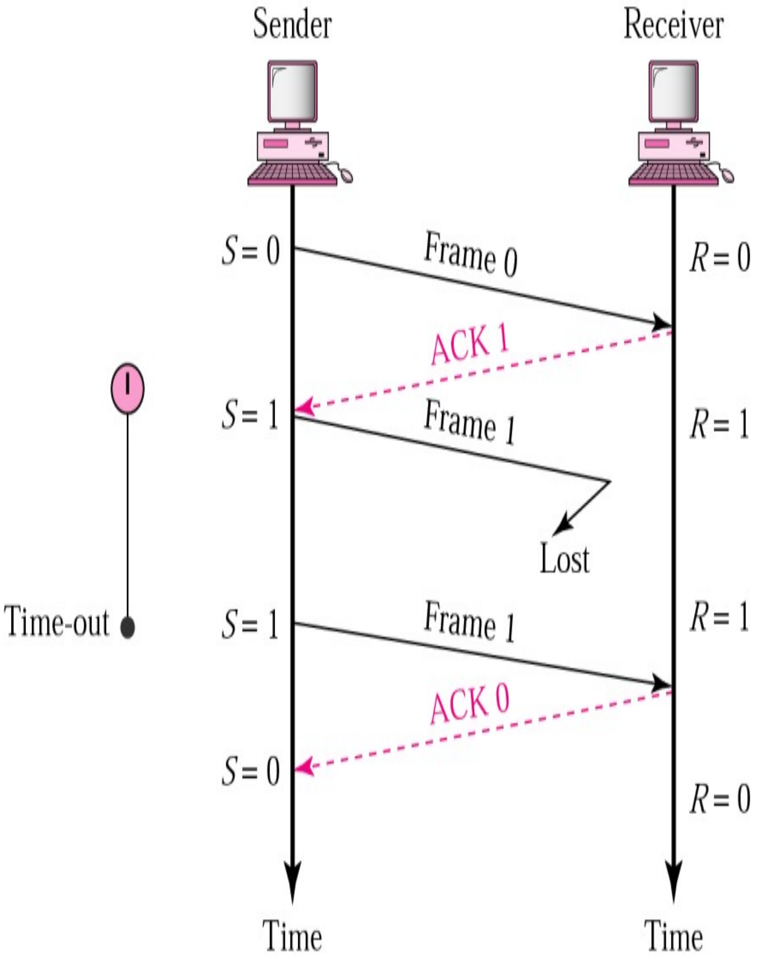
- *The sender will not send the next frame until it is sure that the current one is correctly receive*
- *sequence number is necessary to check for duplicated frames*



# 1. Stop and Wait ARQ

## 2. Lost or damaged frame

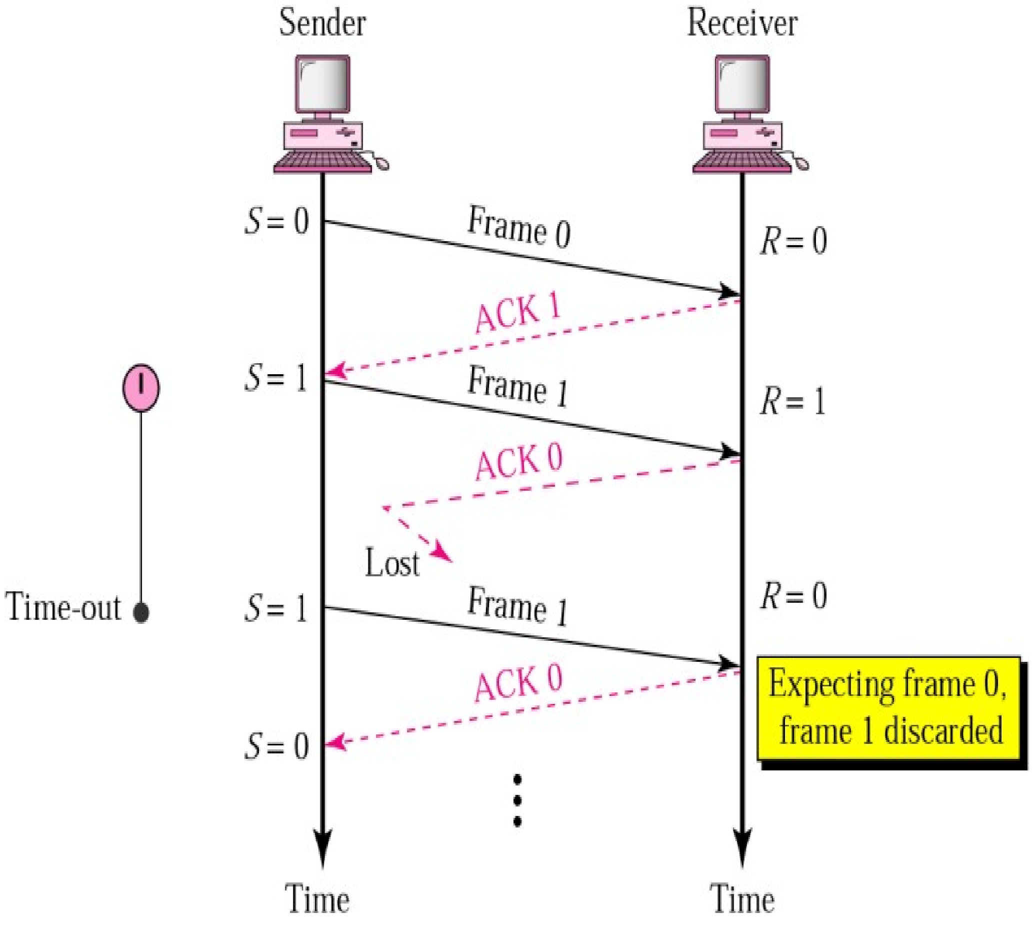
- *A damage or lost frame treated by the same manner by the receiver.*
- *No NACK when frame is corrupted / duplicate*



# *Stop-and-Wait ARQ*

## 3. Lost ACK frame

- *Importance of frame numbering*



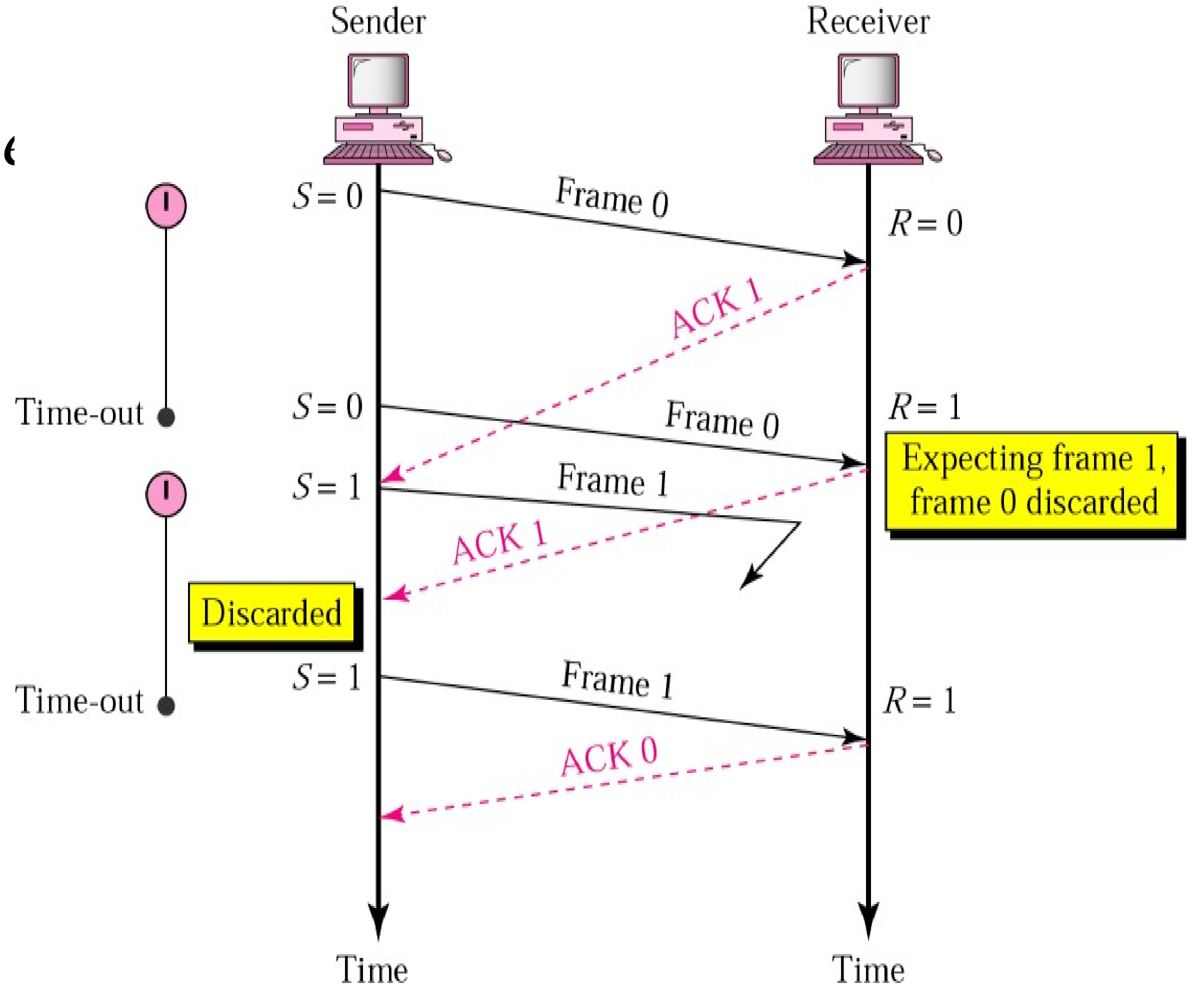
*Note*

***In Stop and-Wait ARQ, numbering frames prevents the retaining of duplicate frames.***

# *Stop-and-Wait ARQ*

## 4. Delayed ACK and lost frame

- *Importance of frame numbering*



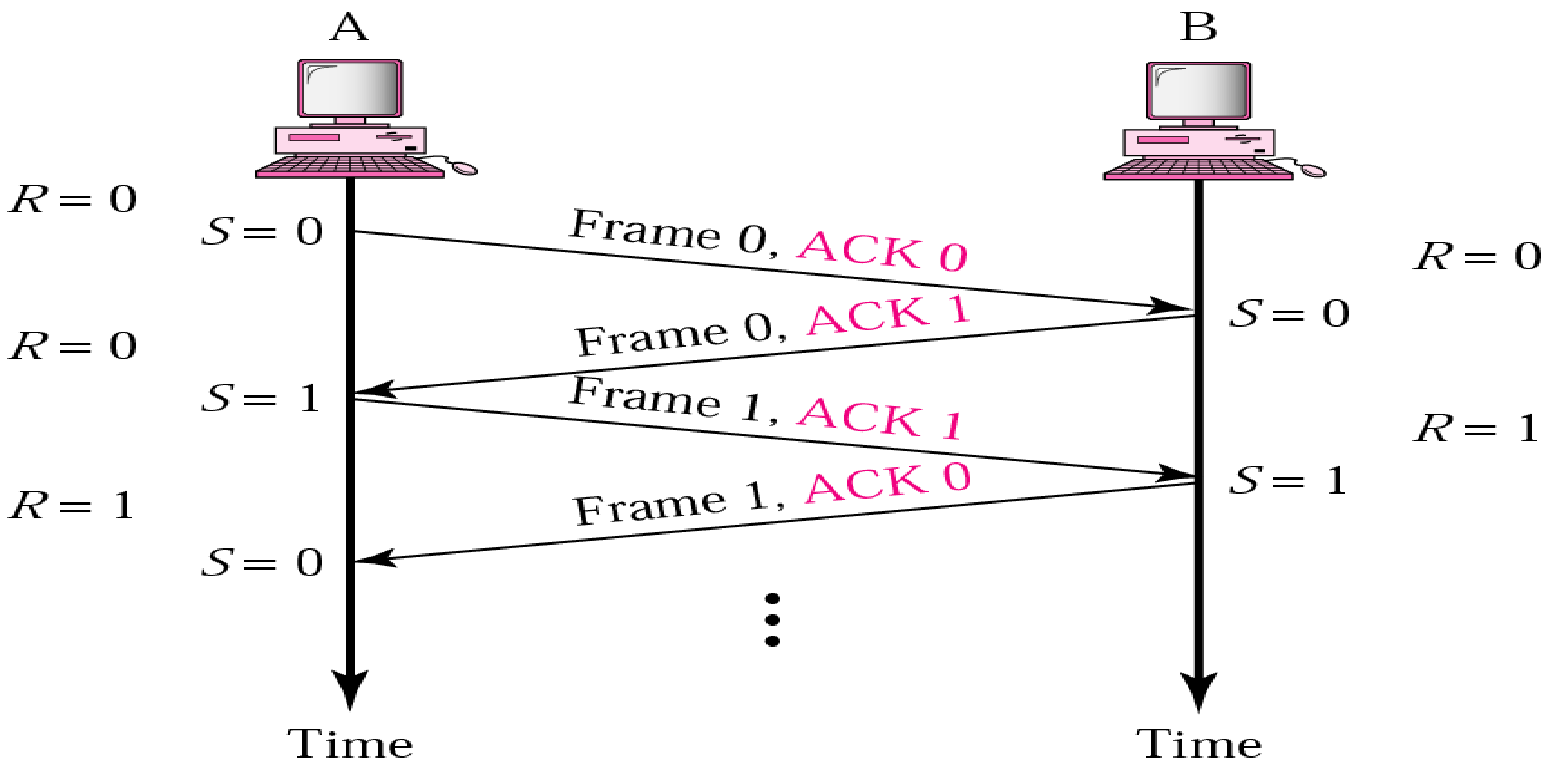
*Note*

***Numbered acknowledgments are needed if an acknowledgment is delayed and the next frame is lost.***

# Piggybacking ( Bidirectional transmission)

*Is a method to combine a data frame with an acknowledgment.*

*It can save bandwidth because data frame and an ACK frame can combined into just one frame*



# *Stop-and-Wait ARQ*

*After each frame sent the host must wait for an ACK*

- *inefficient use of bandwidth*

*To improve efficiency ACK should be sent after multiple frames*

*Alternatives: Sliding Window protocol*

- ✓ *Go-back-N ARQ*
- ✓ *Selective Repeat ARQ*

# Pipelining

*Pipelining: A task is begun before the previous task has ended*

- *There is no pipelining in stop and wait ARQ because we need to wait for a frame to reach the destination and be acknowledged before the next frame can be sent*
- *Pipelining improves the efficiency of the transmission*

# Sliding window protocol

*Sliding window protocols apply Pipelining :*

- ✓ *Go-Back-N ARQ*
- ✓ *Selective Repeat ARQ*
- ➢ *Sliding window protocols improve the efficiency*
- ➢ *multiple frames should be in transition while waiting for ACK. Let more than one frame to be outstanding.*
- ➢ *Outstanding frames: frames sent but not acknowledged*
- ➢ *We can send up to W frames and keep a copy of these frames(outstanding) until the ACKs arrive.*
- ➢ *This procedures requires additional feature to be added :sliding window*

# Sliding window

## Sequence Numbers

- *Sent frames are numbered sequentially*
- *Sequence number is stored in the header of the frame*
- *If the header of the frame allow m bits for the sequence number, the sequence numbers range from 0 to ($2^m$ -1).*

*The sequence numbers are modulo $2^m$ ,where m is the size of the sequence number field in bits.*

*If m = 3, sequence number range from 0 to 7( 8 numbers): 0, 1, 2 , 3, 4, 5, 6, 7, 0 , 1 ,…..*

*Sliding window*

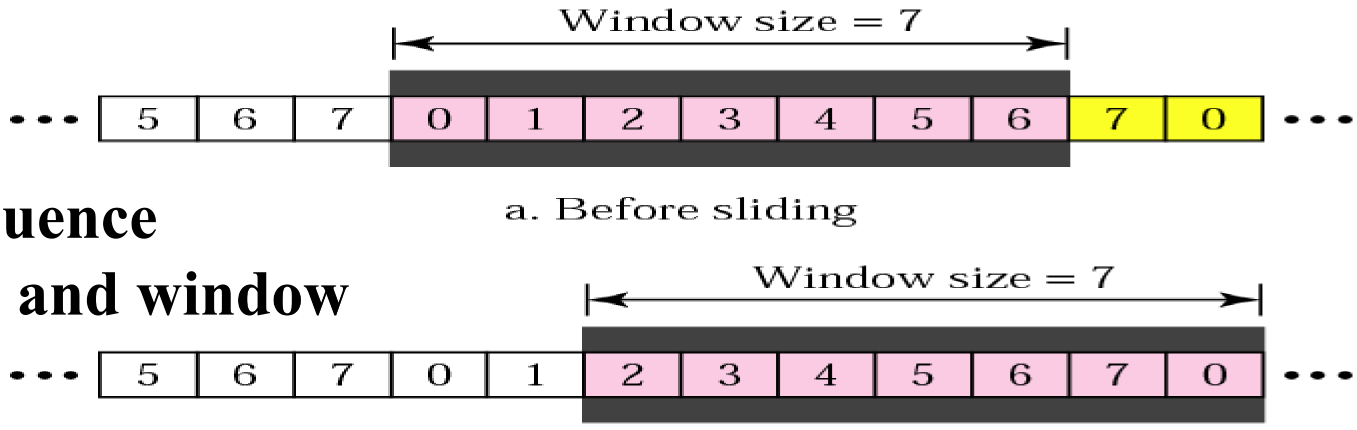
*used to hold the unacknowledged outstanding frames (frames sent but not acknowledged)*

# *Go_Back _N ARQ*

## *Sender sliding window*

*The sender window is an abstract concept defining an imaginary box of size* $\mathbf{2^m} - 1$ *( sequence numbers –1)*
*The sender window can slide one or more slots when a valid acknowledgment arrives.*

**If m = 3; sequence numbers = 8 and window size =7**



a. Before sliding

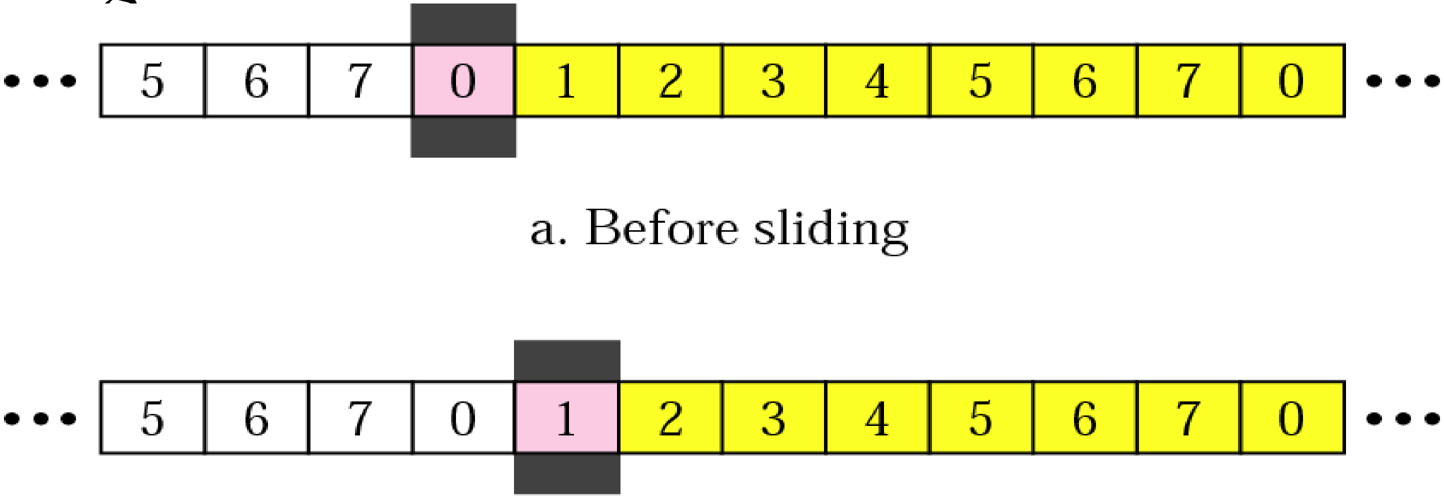
b. After sliding two frames

**Acknowledged frames**

# *Go_Back _N ARQ*

***Receiver sliding window***

- *The receive window is an abstract concept defining an imaginary box of size 1 with one single variable Rn.*
- *The window slides when a correct frame has arrived; sliding occurs one slot at a time.*

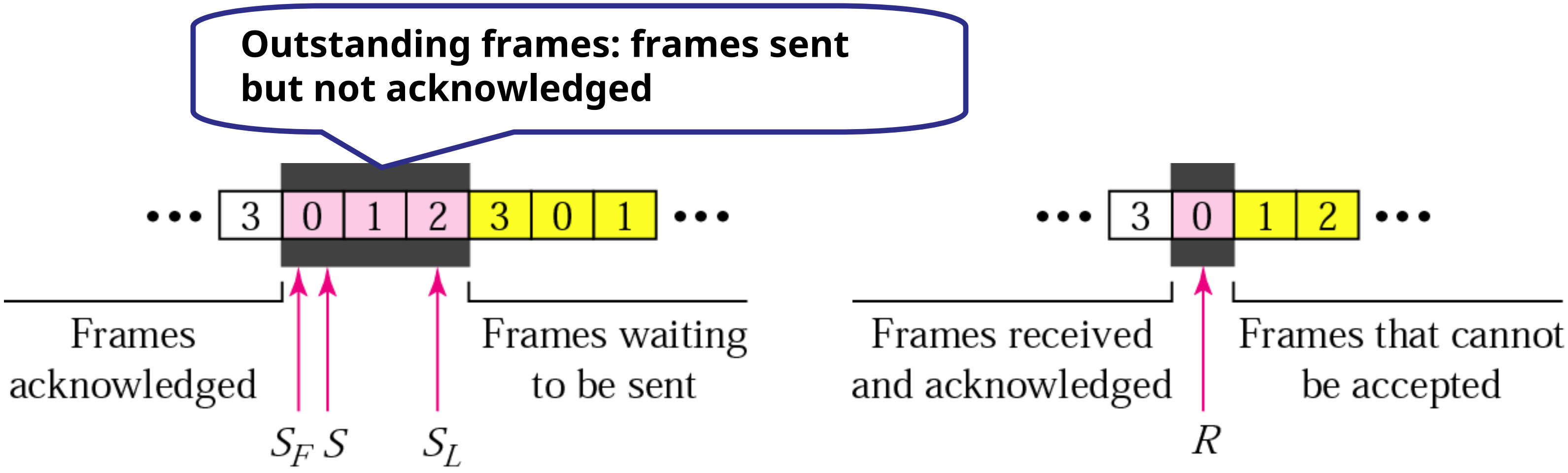


a. Before sliding

b. After sliding

# *Go-Back-N ARQ*

## control variables



**Outstanding frames: frames sent but not acknowledged**

3
0
1
2
3
0
1
Frames acknowledged
Frames waiting to be sent
$S_F$
$S$
$S_L$
3
0
1
2
Frames received and acknowledged
Frames that cannot be accepted
$R$

a. Sender window

b. Receiver window

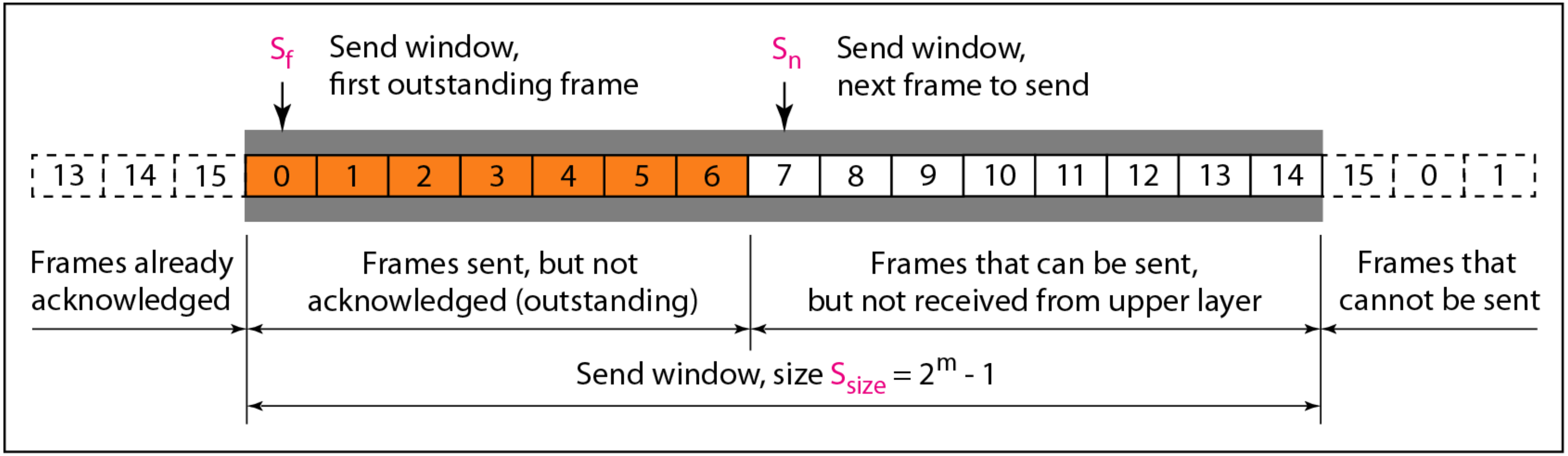
*S: hold the sequence number of the recently sent frame*

*$S_F$: holds sequence number of the first frame in the window*

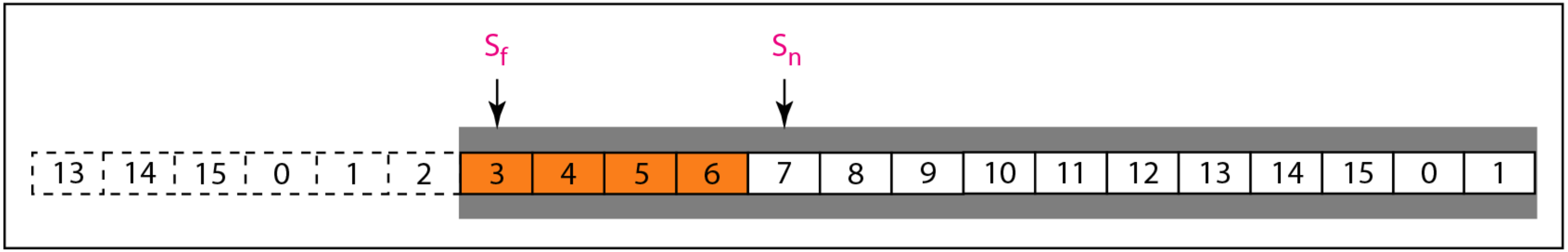
*$S_L$: holds the sequence number of the last frame*

*R: sequence number of the frame expected to received*

# *Go-Back-N ARQ*


$S_f$
Send window,
first outstanding frame
$S_n$
Send window,
next frame to send
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1
Frames already acknowledged
Frames sent, but not acknowledged (outstanding)
Frames that can be sent, but not received from upper layer
Frames that cannot be sent
Send window, size $S_{size} = 2^m - 1$

a. Send window before sliding



b. Send window after sliding

# ***Go-Back-N ARQ***

*In Go-Back-N ARQ we use one timer for the first outstanding frame*

- ➢*The receiver sends a positive ACK if a frame has arrived safe and in order.*
- ➢*if a frame is damaged or out of order ,the receiver is silent and will discard all subsequent frames*
- ➢*When the timer of an unacknowledged frame at the sender site is expired , the sender goes back and resend all frames , beginning with the one with expired timer.( that is why the protocol is called Go-Back-N ARQ)*
- ➢*The receiver doesn't have to acknowledge each frame received . It can send cumulative Ack for several frame*

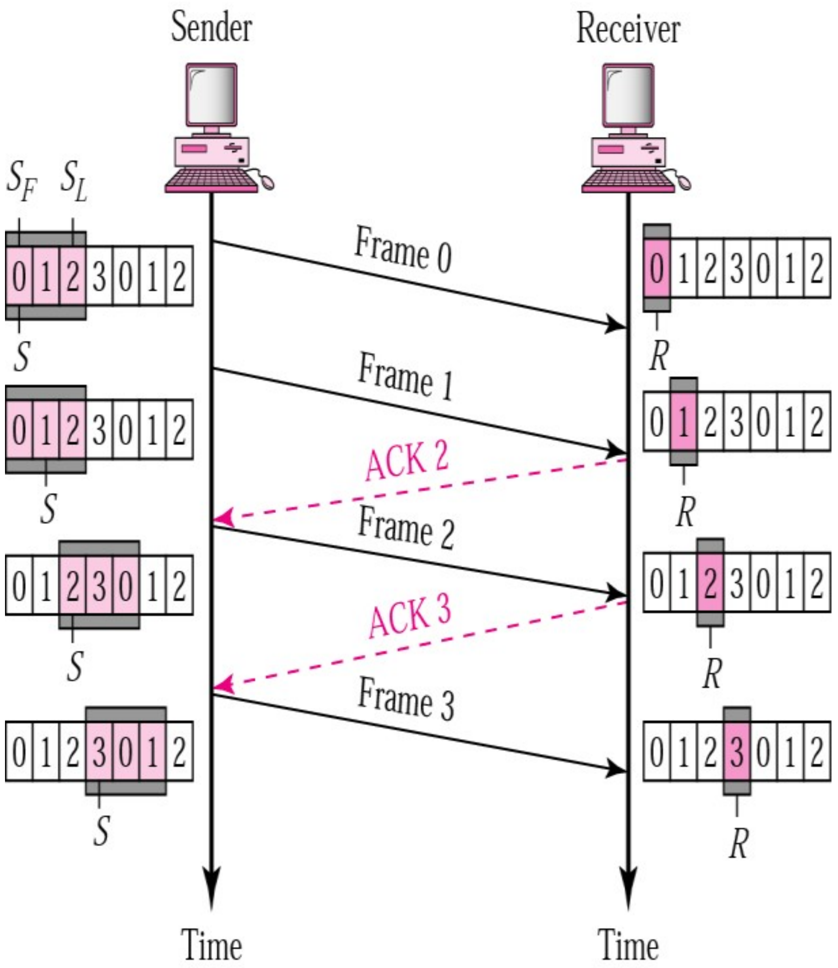
# ***Go-Back-N ARQ***

*Example: The sender has sent frame 6 , and timer expires for frame 3( frame 3 has not been acknowledge); the sender goes back and resends frames 3, 4,5 and 6*

# *Go-Back-N ARQ*

## Normal operation

- *How many frame scan be transmitted Without acknowledgment?*
- *ACK1 is not necessary if ACK2 is sent: Cumulative ACK*

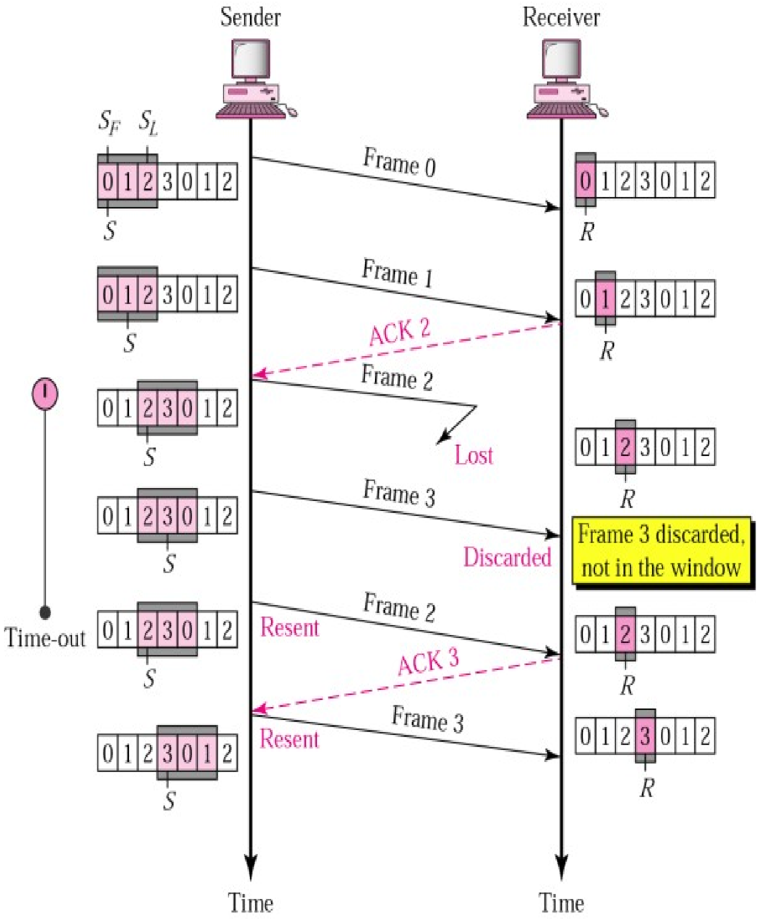


# Go-Back-N ARQ

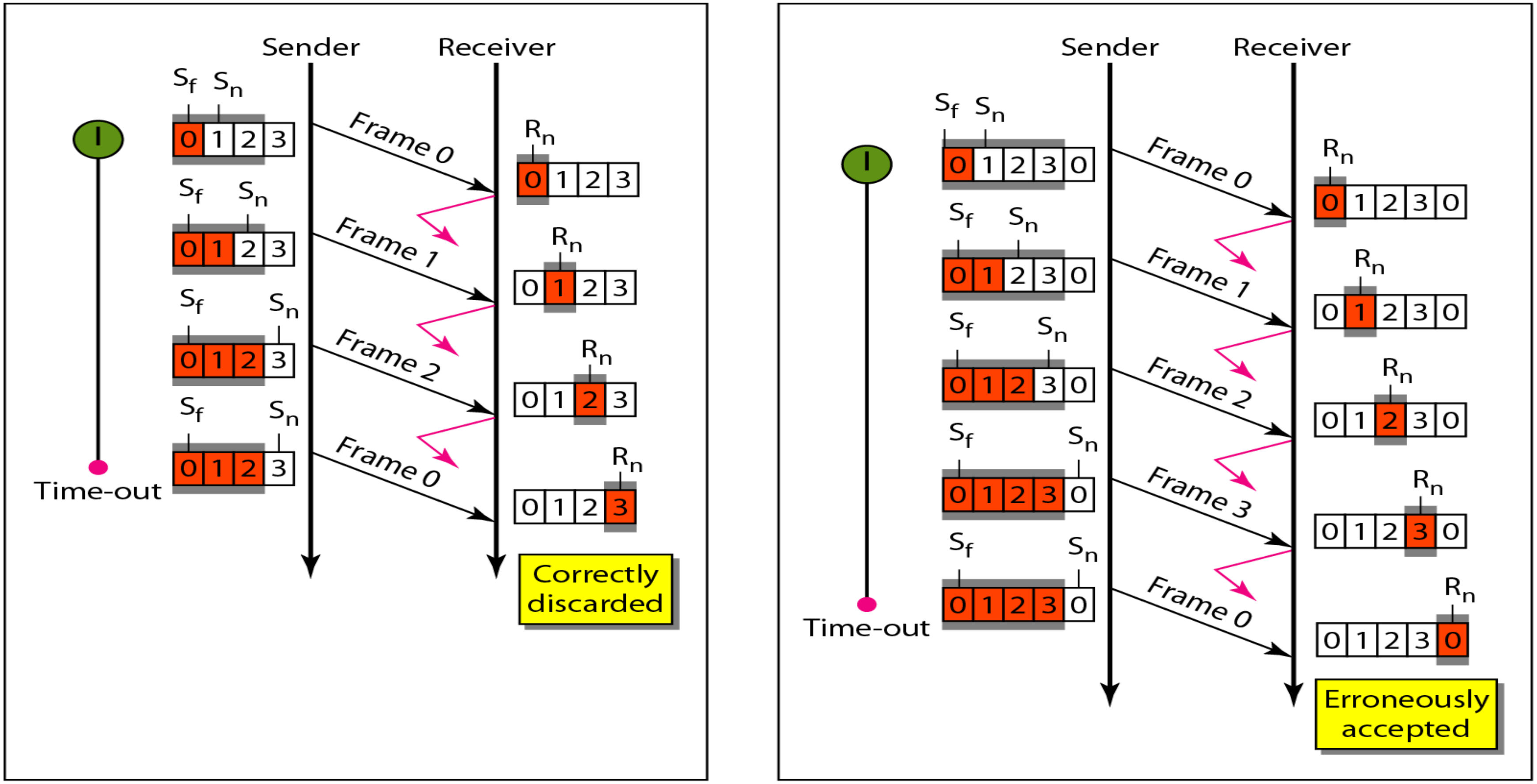
## Damage or Lost Frame

*Correctly received out of order packets are not Buffered*

*What is the disadvantage of this?*



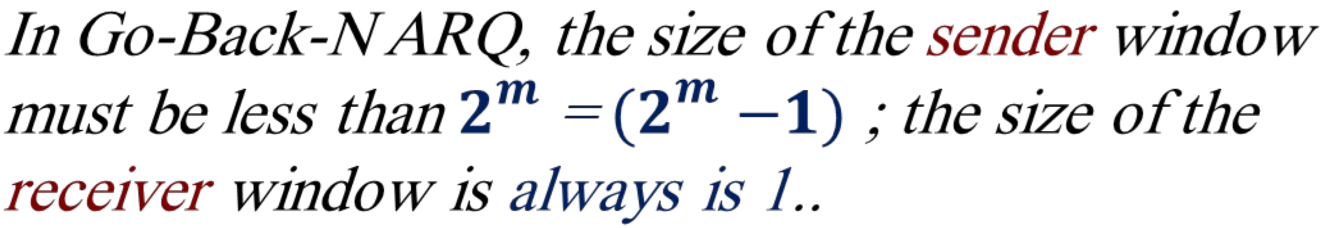
# Go-Back-N ARQ



a. Window size < $2^m$

b. Window size = $2^m$

# Go-Back-N ARQ



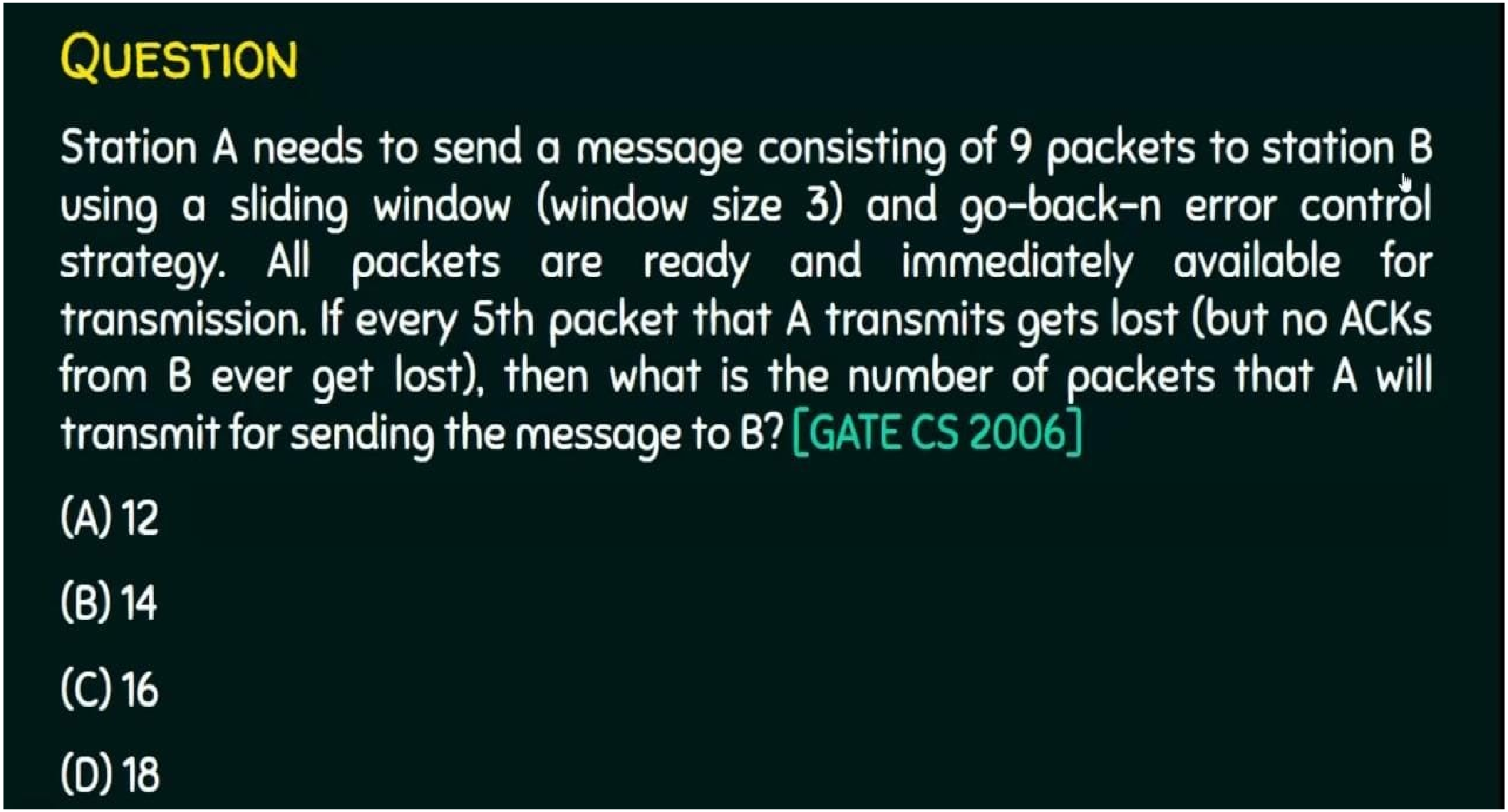
*In Go-Back-N ARQ, the size of the sender window must be less than* $\mathbf{2^m} = \mathbf{(2^m - 1)}$ *; the size of the receiver window is always is 1..*

***Bidirectional transmission : piggybacking***

*As Stop-and-Wait we can use piggybacking to improve the efficiency of bidirectional transmission . Each direction needs both a sender window and a receiver window.*

**Note**

*Stop-and-Wait ARQ is a special case of Go-Back-N ARQ in which the size of the send window is 1*



## QUESTION

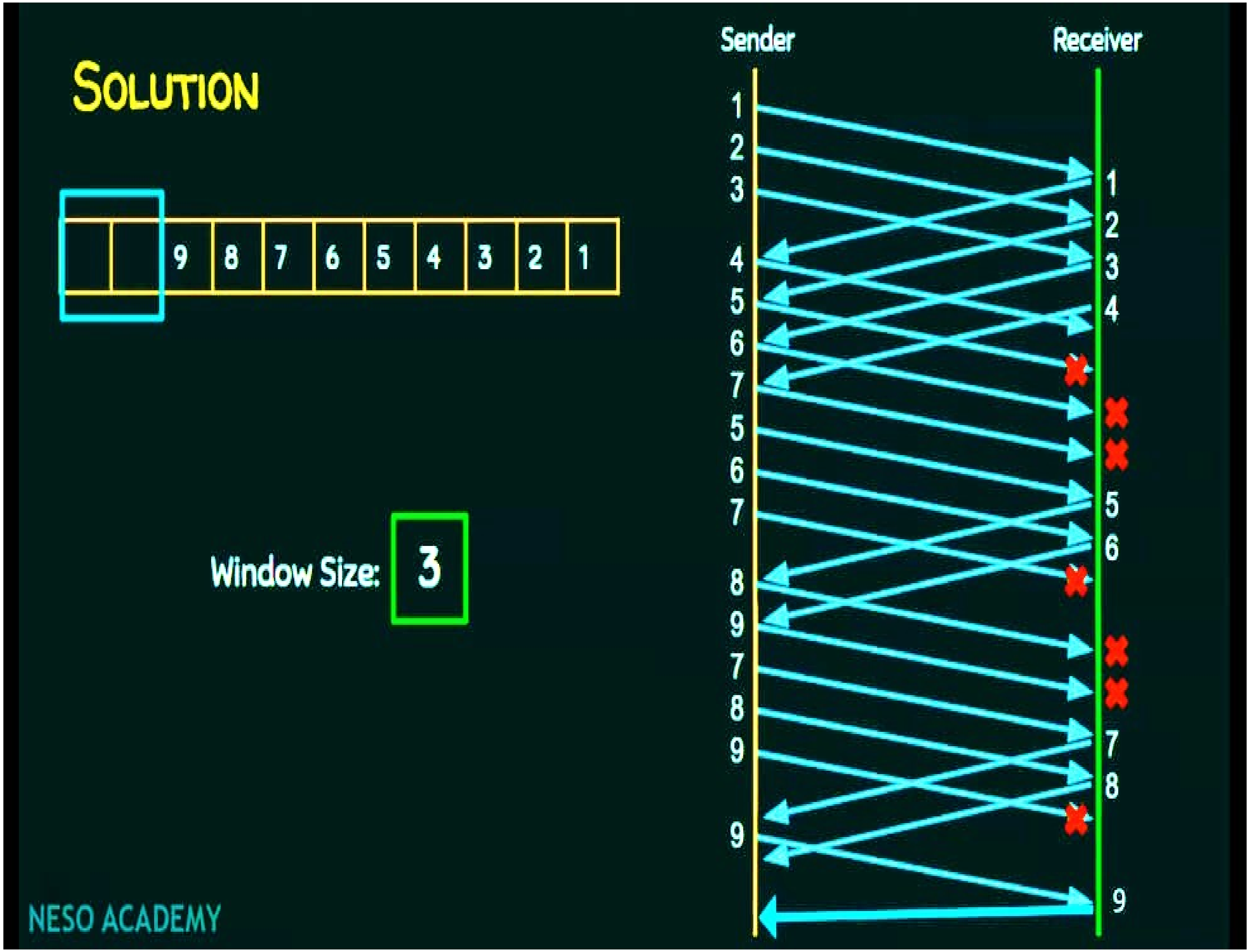
Station A needs to send a message consisting of 9 packets to station B using a sliding window (window size 3) and go-back-n error control strategy. All packets are ready and immediately available for transmission. If every 5th packet that A transmits gets lost (but no ACKs from B ever get lost), then what is the number of packets that A will transmit for sending the message to B? [GATE CS 2006]

(A) 12

(B) 14

(C) 16

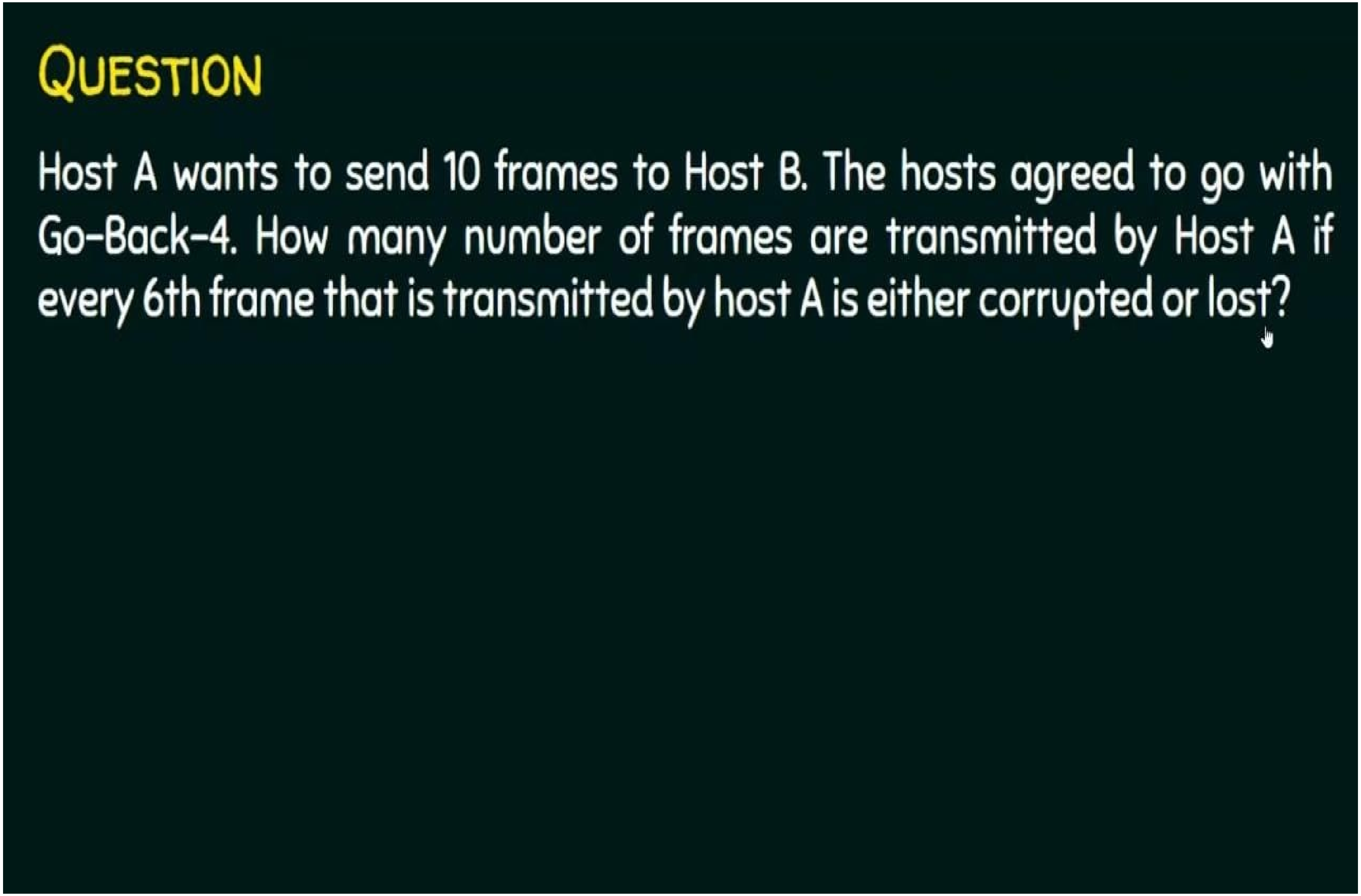
(D) 18



# SOLUTION

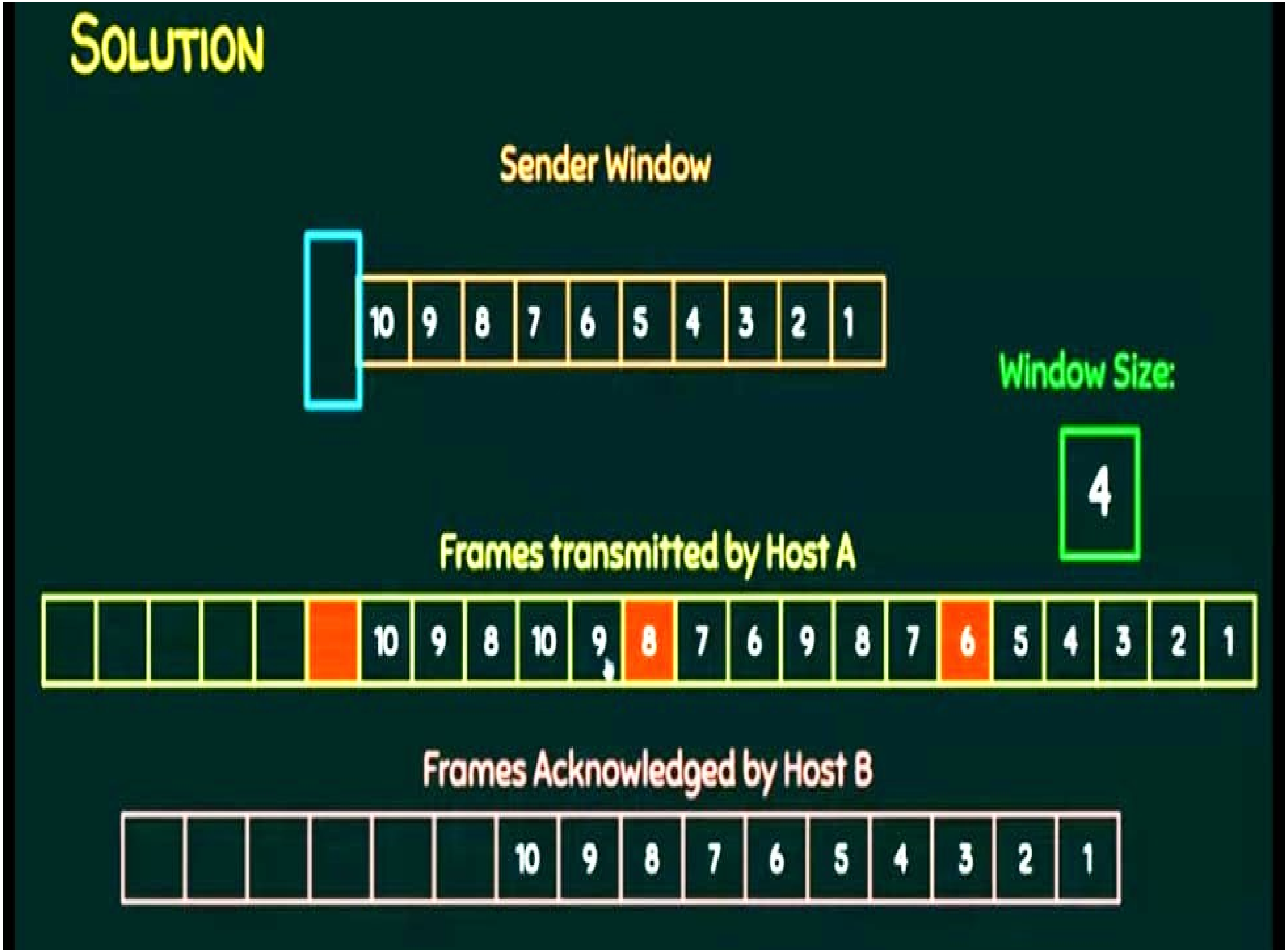
Window Size: 3

NESO ACADEMY



## QUESTION

Host A wants to send 10 frames to Host B. The hosts agreed to go with Go-Back-4. How many number of frames are transmitted by Host A if every 6th frame that is transmitted by host A is either corrupted or lost?



# Solution

Sender Window

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |

Window Size:

4

Frames transmitted by Host A

| | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | 🟥 | 10 | 9 | 8 | 10 | 9 | 8 | 7 | 6 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |

Frames Acknowledged by Host B

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |

# *Selective Repeat ARQ*

*Go-Back-N ARQ is inefficient of a* ***noisy*** *link.*

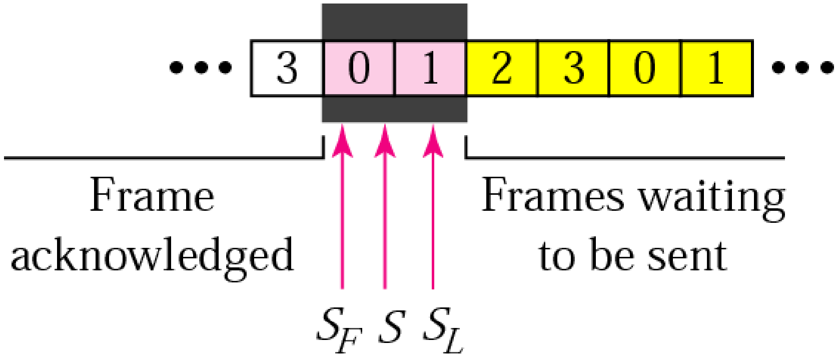
- *In a noisy link frames have higher probability of damage , which means the resending of multiple frames.*
- *this resending consumes the bandwidth and slow down the transmission .*

***Solution***:

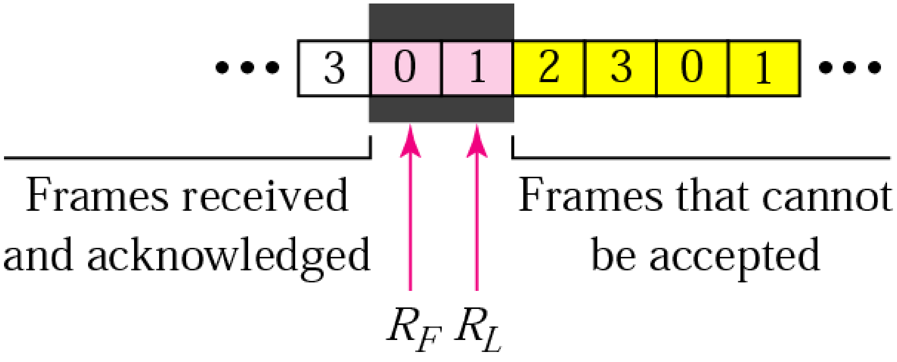
- *Selective Repeat ARQ protocol : resent only the damage frame*
- *It defines a negative Acknolgment (NAK) that report the sequence number of a damaged frame before the timer expires*
- *It is more efficient for noisy link, but the processing at the receiver is more complex*

# *Selective Repeat ARQ*

- *The window size is reduced to one half of* $\mathbf{2^m}$
- *Sender window size = receiver window size =* $\mathbf{2^m}$ */2*
- *Window size = sequence number/2*
- *If m = 2, Window size = 4/2=2*
- *Sequence number = 0, 1, 2 , 3*



a. Sender window



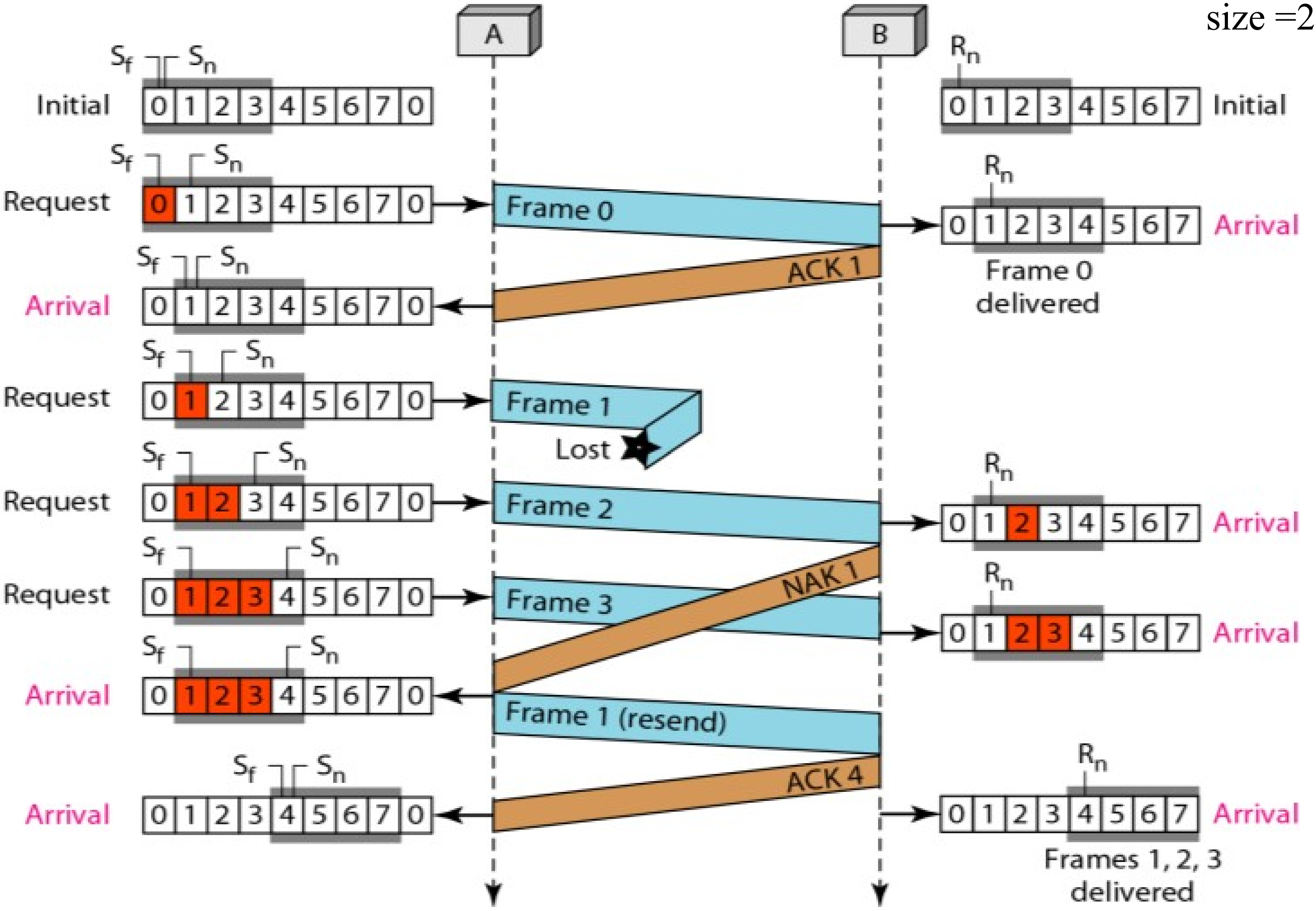
3
0
1
2
3
0
1
Frames received and acknowledged
Frames that cannot be accepted
$R_F$ $R_L$

b. Receiver window

# *Selective Repeat ARQ*

## Lost Frame

m=3
Sequences no=2m =8 :
0,1,2 ,3,4,5,6,7Window
size =2m/2= 8/2=4



# Selective Repeat ARQ

*At the receiver site we need to distinguish between the acceptance of a frame and its delivery to the network layer . At the second arrival , frame 2 arrives and is stored and marked , but it can not be delivered because frame 1 is missing . At the next arrival , frame 3 arrives and is marked and stored , but still none of the frames can be delivered . Only at the last arrival , when finally a copy of frame 1 arrives , can frames 1 , 2 , and 3 be delivered to the network layer. There are two conditions for the delivery of frames to the network layer: First , a set of consecutive frames must have arrived. Second, the set starts from the beginning of the window* **.**

# *Selective Repeat ARQ*

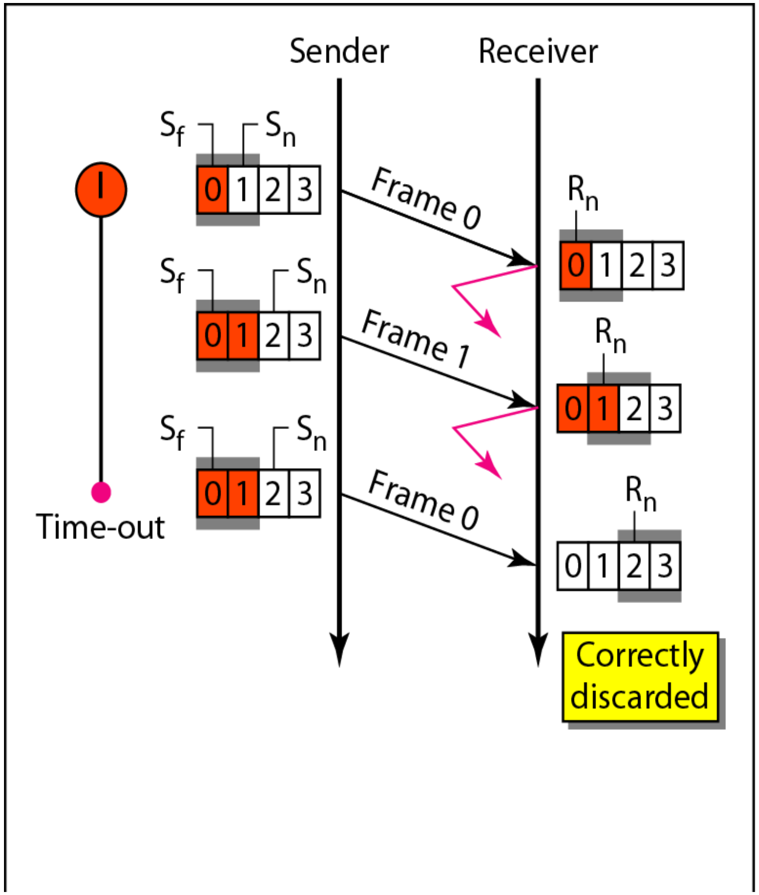
*The next point is about the ACKs . Notice that only two ACKs are sent here. The first one acknowledges only the first frame; the second one acknowledges three frames. In Selective Repeat, ACKs are sent when data are delivered to the network layer. If the data belonging to n frames are delivered in one shot , only one ACK is sent for all of them.*

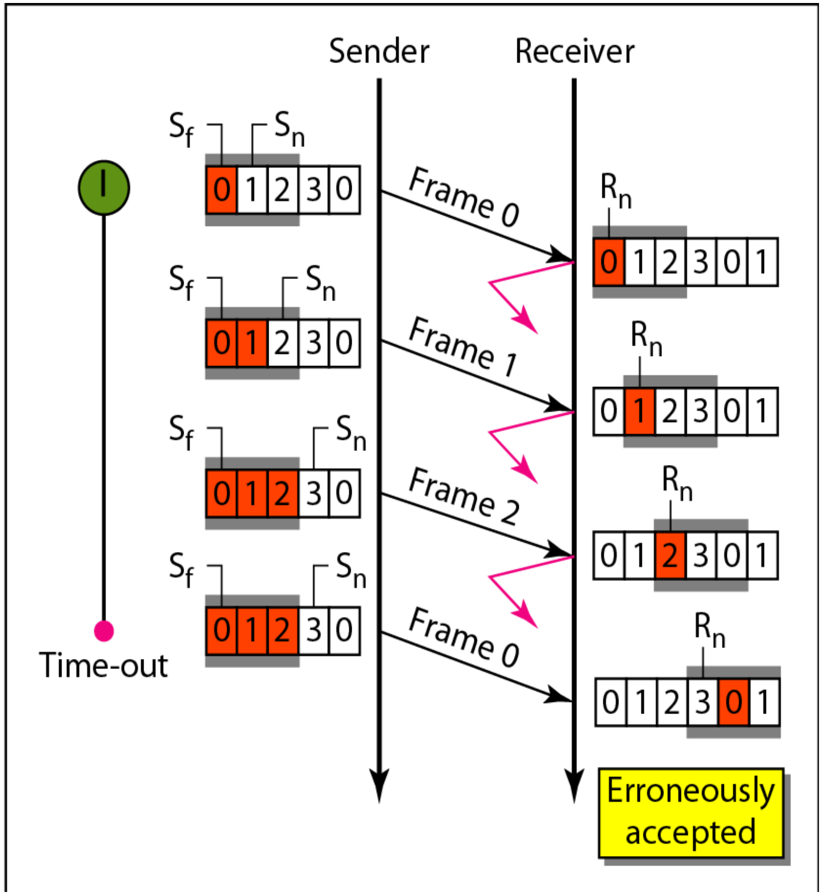
**Note**

*In Selective Repeat ARQ, the size of In the sender and receiver window must be at most one-half of 2m.*

# *Selective Repeat ARQ*

**m=2**


Sender
Receiver
$S_f$
$S_n$
0 1 2 3
Frame 0
$R_n$
0 1 2 3
$S_f$
$S_n$
0 1 2 3
Frame 1
$R_n$
0 1 2 3
$S_f$
$S_n$
0 1 2 3
Frame 0
$R_n$
0 1 2 3
Time-out
Correctly discarded


Sender
Receiver
$S_f$
$S_n$
0 1 2 3 0
Frame 0
$R_n$
0 1 2 3 0 1
$S_f$
$S_n$
0 1 2 3 0
Frame 1
$R_n$
0 1 2 3 0 1
$S_f$
$S_n$
0 1 2 3 0
Frame 2
$R_n$
0 1 2 3 0 1
$S_f$
$S_n$
0 1 2 3 0
Frame 0
$R_n$
0 1 2 3 0 1
Time-out
Erroneously accepted