# Data Communication & Networks

Chapter 6: Data Link Layer (Flow Control)

Chapter 11 of Reference Book

# Flow Control

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

  - The flow of data must not be allowed to overwhelm the receiver.

# Error Control

- Error control is both error detection and error correction.

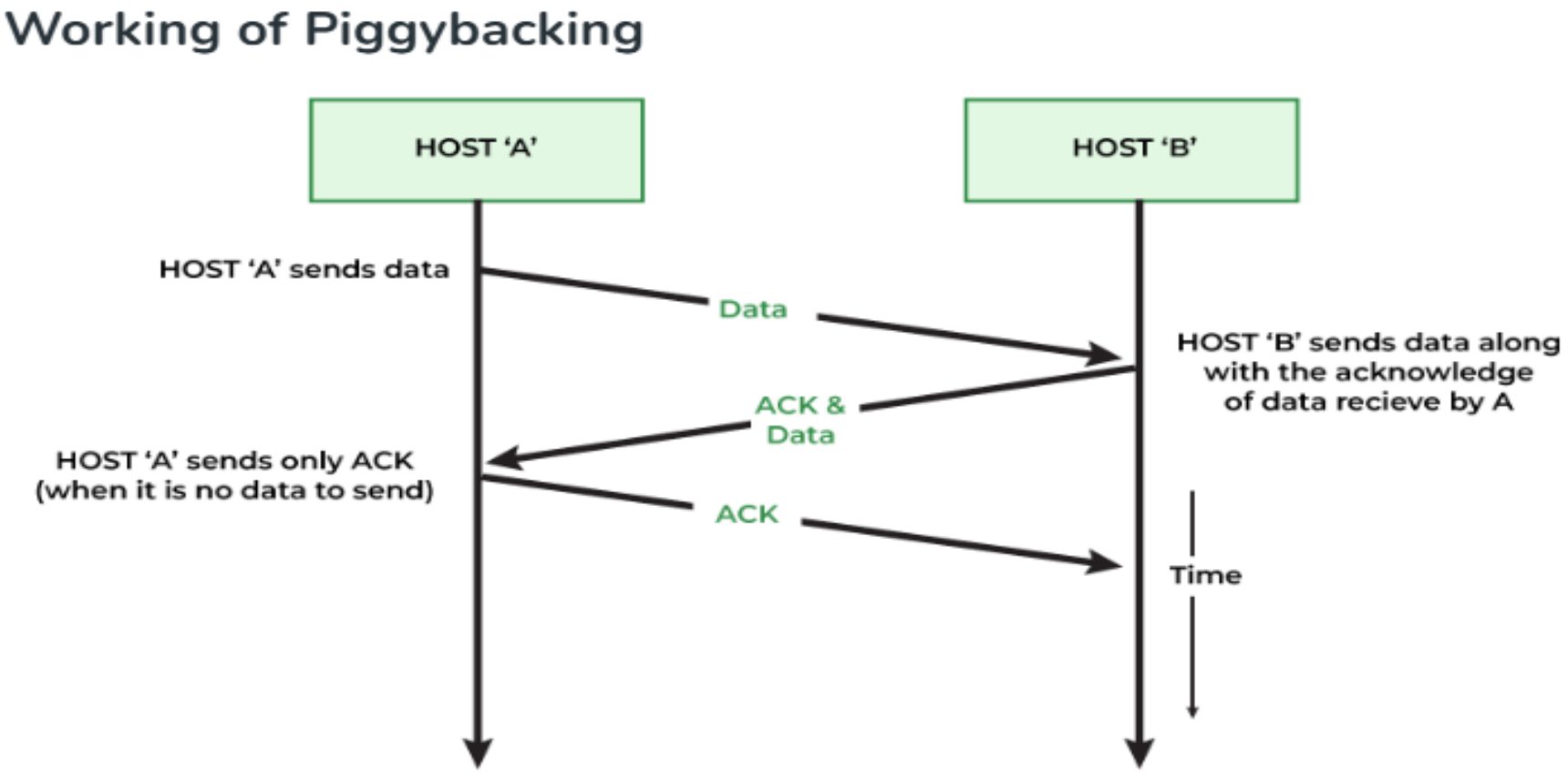
- Error control is often implemented simply by Automatic Repeat Request (ARQ):

1. Any time an error is detected in an exchange, Receiver sends the message to transmitter.
2. Transmitter retransmits the specified frames.

# Data Link Layer Protocols: ARQ Protocols

- Acknowledgement (ACK) Frame.

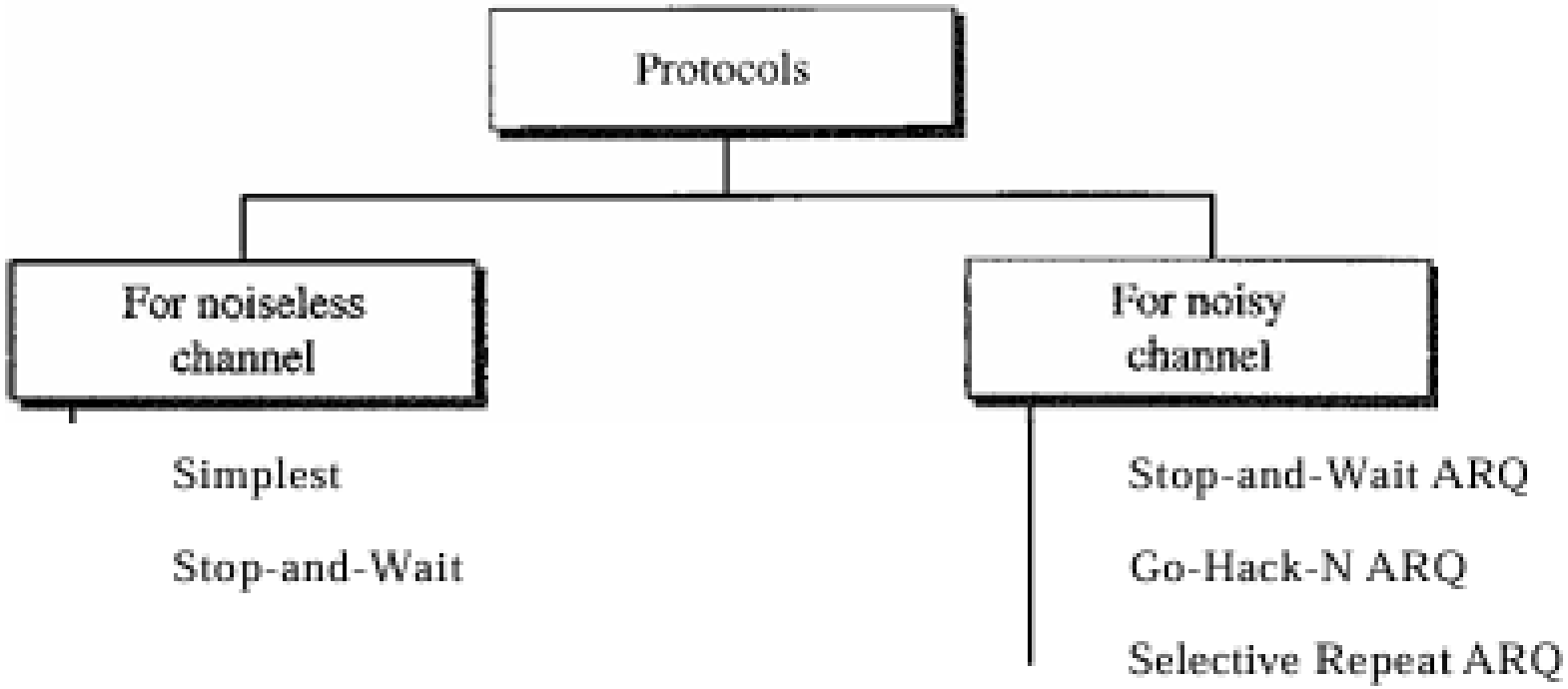
- Negative Acknowledgement (NAK) Frame




*Working of Piggybacking*

# ARQ Protocols

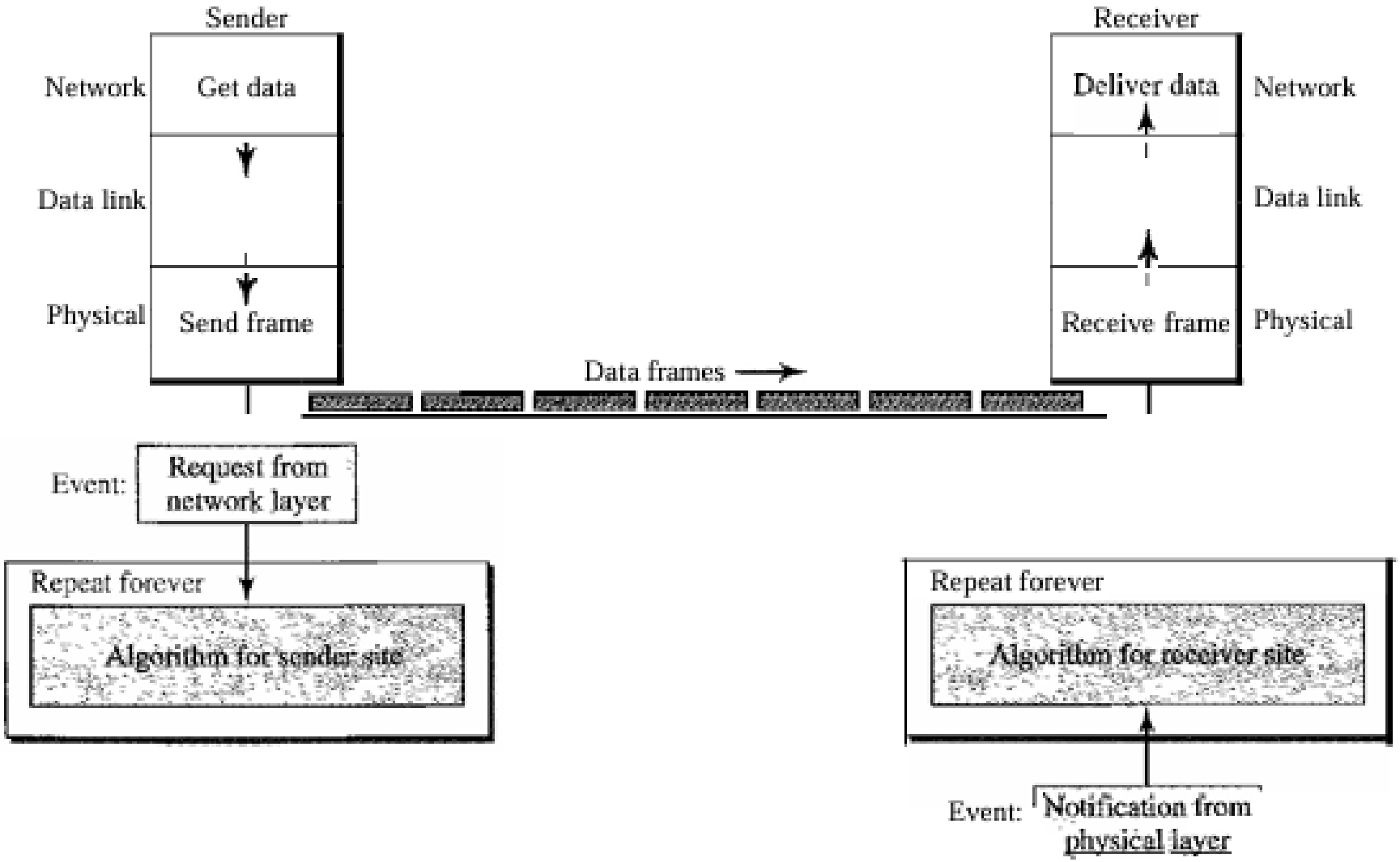
Figure 11.5 *Taxonomy ofprotocols discussed in this chapter*




- The protocols in the first category cannot be used in real life. Unidirectional
- They are the basis of noisy channel ARQ protocol. Bidirectional

# Noiseless Channel: Simplest Protocol

Figure 11.6 *The design of the simplest protocol with no flow or error control*



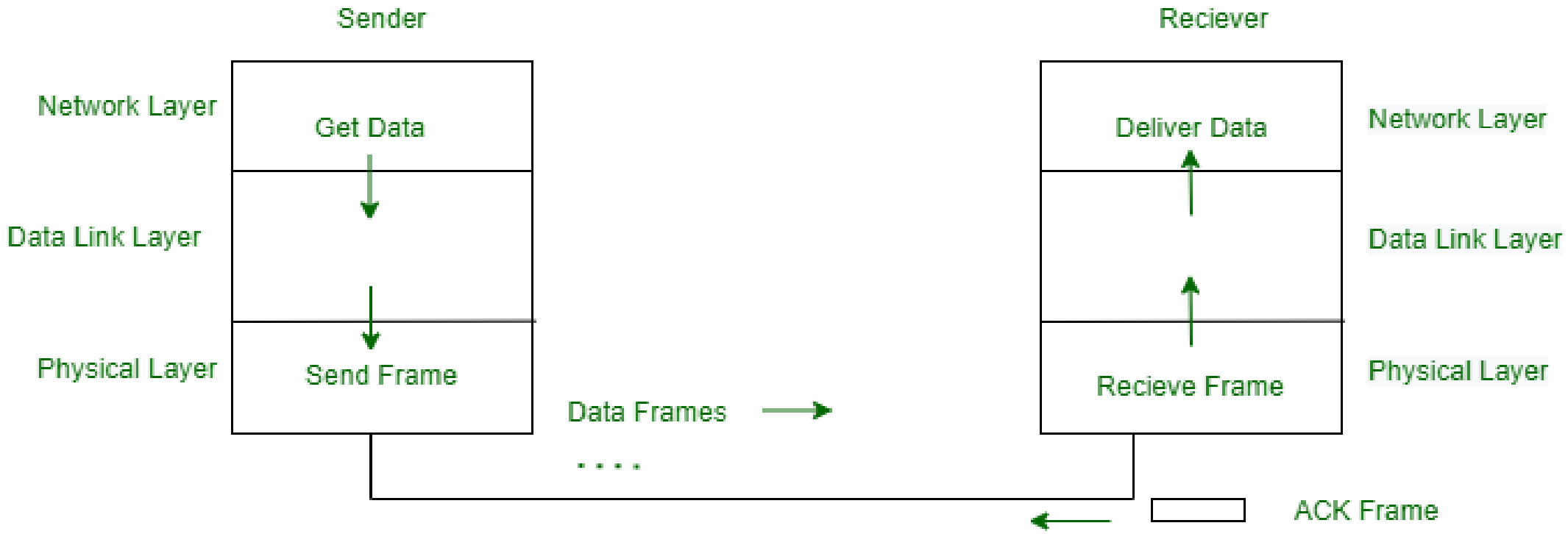
# Noiseless Channel: Simplest Protocol

| Sender-Side Algorithm | Receiver-Side Algorithm |
| --- | --- |
| **while(true) //Repeat forever**<br>**{**<br>**waitForEvent(); //sleep until an event occur**<br>**if (Event(RequestToSend)) //there is a packet to send**<br>**{**<br>**GetData();**<br>**MakeFrame();**<br>**SendFrame(); //send the frame**<br>**}**<br>**}** | while(true) //Repeat forever<br>{<br>waitForEvent(); //sleep until an event occur<br>if (Event(ArrivalNotification)) //data frame arrived<br>{<br>ReceiveFrame();<br>ExtractData();<br>DeliverData(); //Deliver data to network layer<br>}<br>} |

# Noiseless Channel: Stop-and-Wait Protocol

When data frames arrive at the receiver faster than they can be processed, the receiver must store these frames temporarily until they can be used.

- Typically, receivers have limited storage capacity, especially when receiving data from multiple sources.
- This requires careful management of the data flow to prevent overload.



# Noiseless Channel: Stop-and-Wait Protocol

Algorithm 11.3 *Sender-site algorithm for Stop-and- Wait Protocol*

```
1  while (true)                          //Repeat forever
2  canSend = true                        //Allow the first frame to go
3  {
4    WaitForEvent();                     // Sleep until an event occurs
5    if (Event (RequestToSend) AND canSend)
6    {
7      GetData();
8      MakeFrame();
9      SendFrame();                      //Send the data frame
10     canSend = false;                  //cannot send until ACK arrives
11   }
12   WaitForEvent();                     // Sleep until an event occurs
13   if (Event (ArrivalNotification)     // An ACK has arrived
14    {
15     ReceiveFrame();                   //Receive the ACK frame
16     canSend = true;
17    }
18 }
```

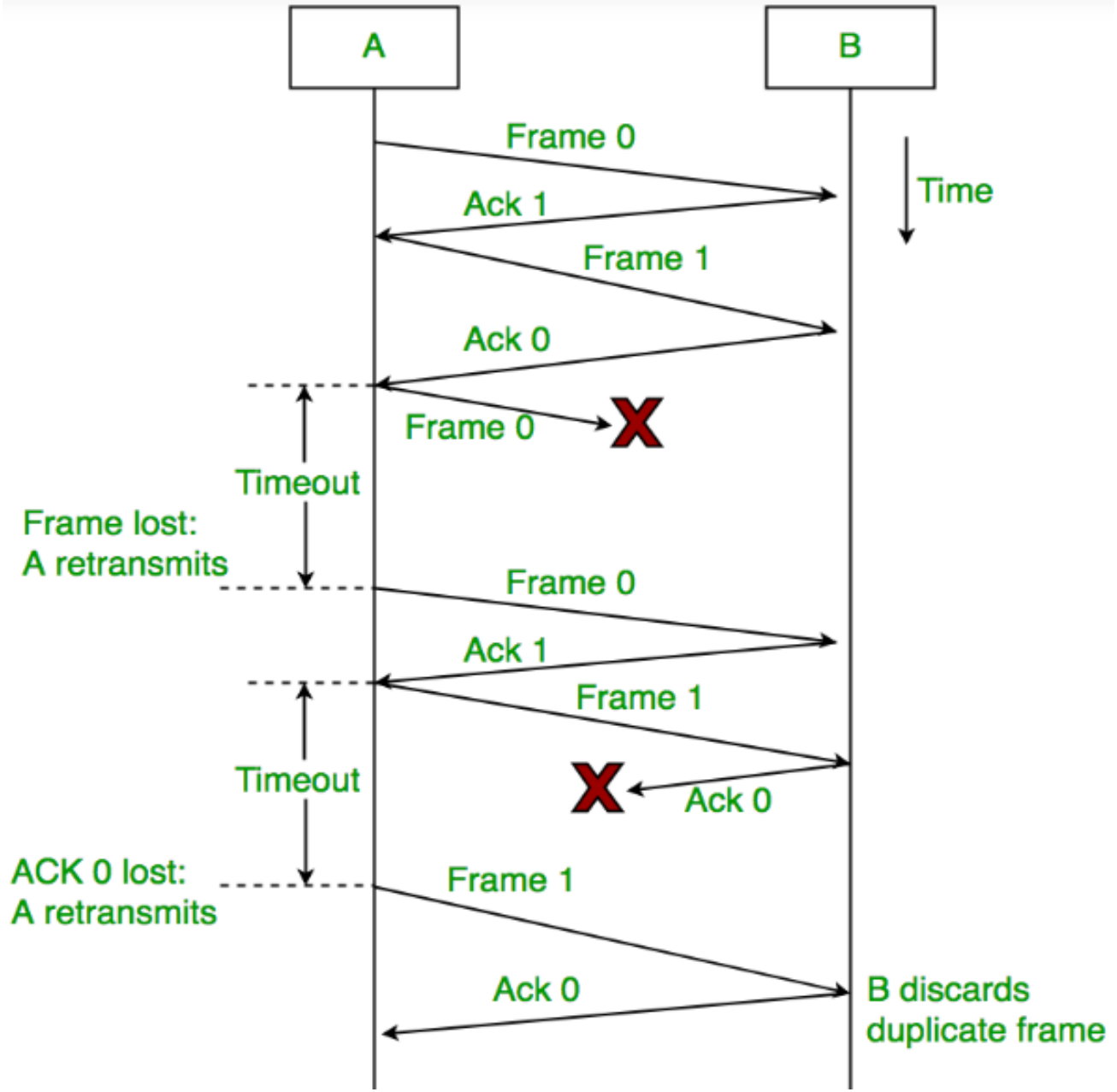
# Noisy Channel: Stop-and-Wait Protocol

Error correction in Stop-and-Wait ARQ is done by keeping a copy of the sent frame and retransmitting of the frame when the timer expires.

1. The frame arrives safe and sound at the receiver site; the receiver sends an acknowledgment. The acknowledgment arrives at the sender site, causing the sender to send the next frame numbered $x + 1$.
2. The frame arrives safe and sound at the receiver site; the receiver sends an acknowledgment, but the acknowledgment is corrupted or lost. The sender resends the frame (numbered *x)* after the time-out. Note that the frame here is a duplicate. The receiver can recognize this fact because it expects frame $x + 1$ but frame $x$ was received.
3. The frame is corrupted or never arrives at the receiver site; the sender resends the frame (numbered *x)* after the time-out.

- The frame numbers are in 0 and 1 (modulo 2).

# Noisy Channel: Stop-and-Wait Protocol



# Sliding Window Protocol

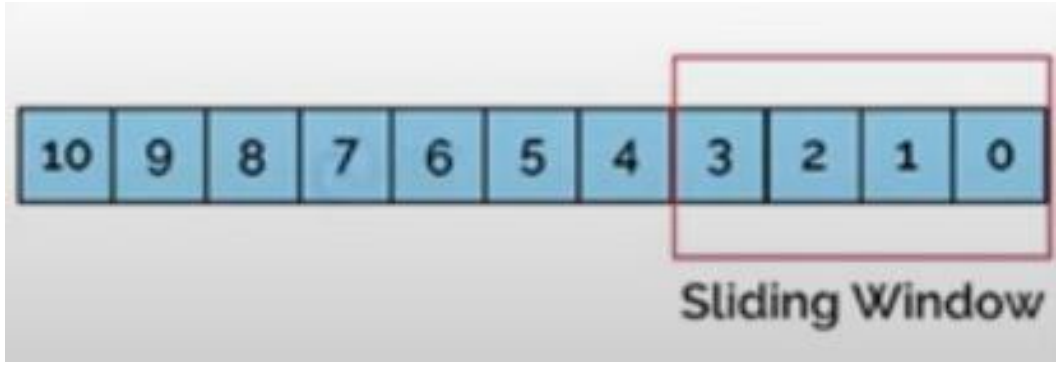
- These are data link layer protocols for reliable and sequential delivery of data frames.
- It is also used in Transmission Control Protocol (TCP).
- This technique is for sending multiple frames at a time to the receiver.
- Each frame has sent from the sequence number.
- The sequence numbers are used to find the missing data in the receiver end.
- The main purpose is to avoid duplicate data, so it uses the sequence number.

## ➢Types of Sliding Window Protocol:

1. Go-Back-N ARQ
2. Selective Repeat ARQ

# Go-Back-N ARQ Protocol

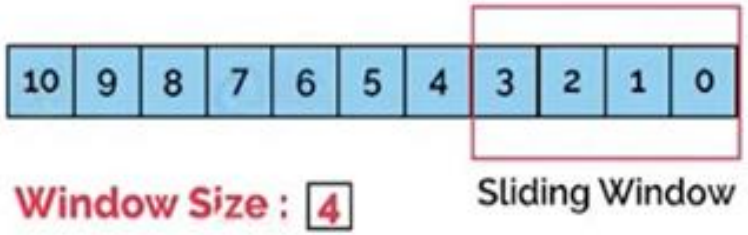
- Go Back N Automatic Repeat Request (ARQ) is a data link layer protocol.
- It is used for data flow control purposes.
- In which multiple frames are sent from sender to receiver at once.
- The sender window is a fixed-sized window 'N' that defines the number of frames that are transmitted from sender to receiver at once.
- The Receiver window in the Go Back N ARQ protocol is always of size 1.
- Receiver takes at most 1 frame at a single time.
- It reject corrupted frames & avoid out of sequence frames.

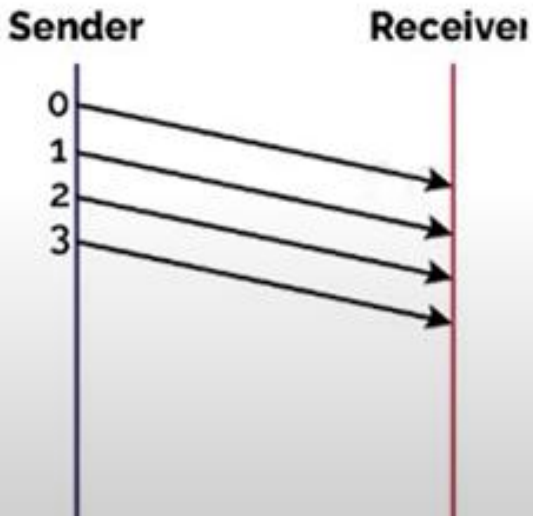



- M-bits originate $2^m$ -1 sequence numbers

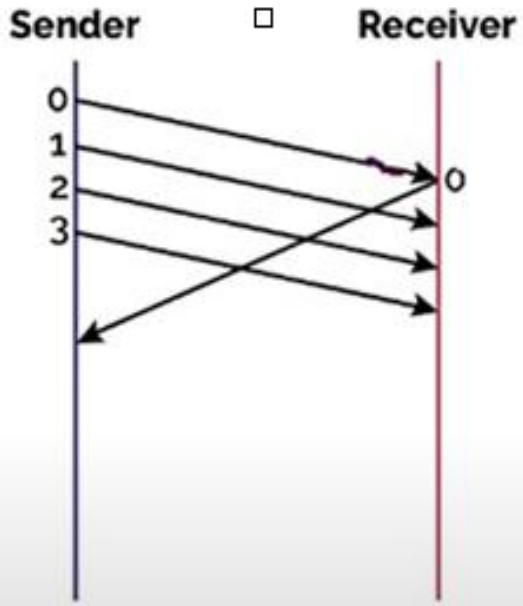
# Go-Back-N ARQ Protocol

➢ **Step 1:** Sender will send first four frames to the receiver (0,1,2,3). N = 4.



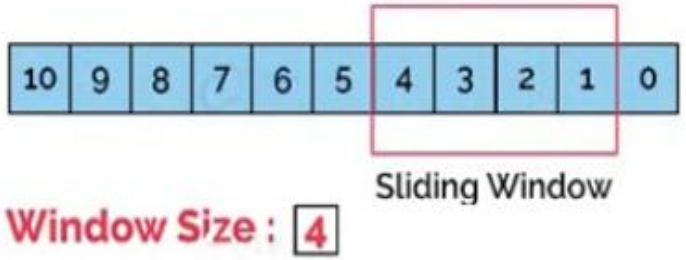



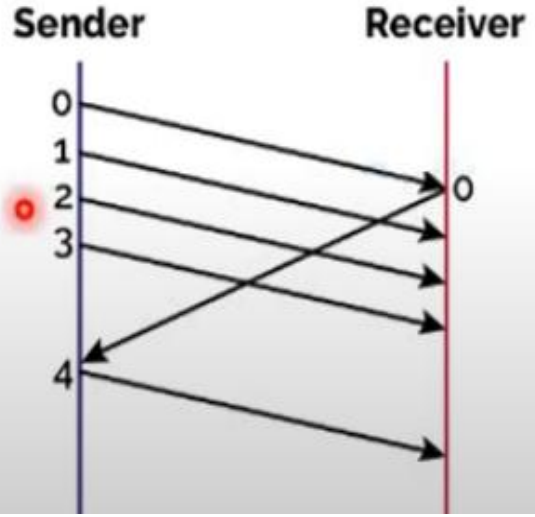

➢ **Step 2:** Sender expects to get ack from 0th frame.



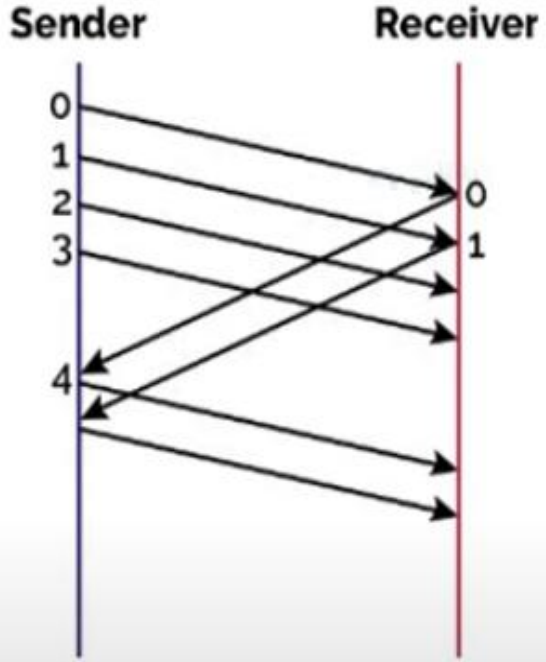
# Go-Back-N ARQ Protocol

➢ **Step 3:** Sender will send following frame, which is 4. The window slides containing four frames (1,2,3,4).



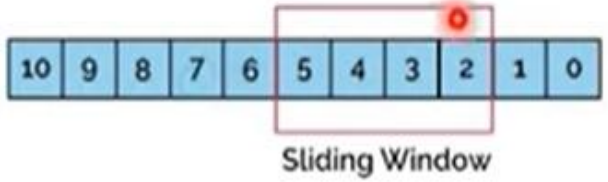




➢ **Step 4:** The acknowledgment for frame no. 1 will then be sent by the receiver.

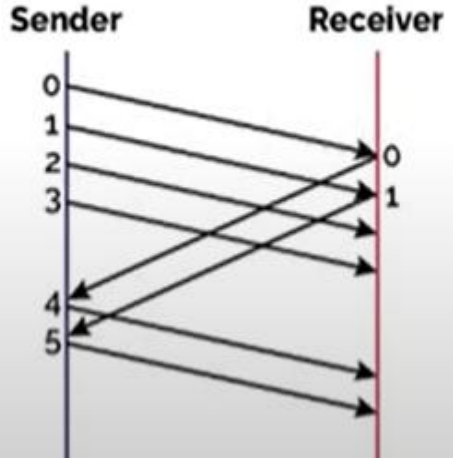


# Go-Back-N ARQ Protocol

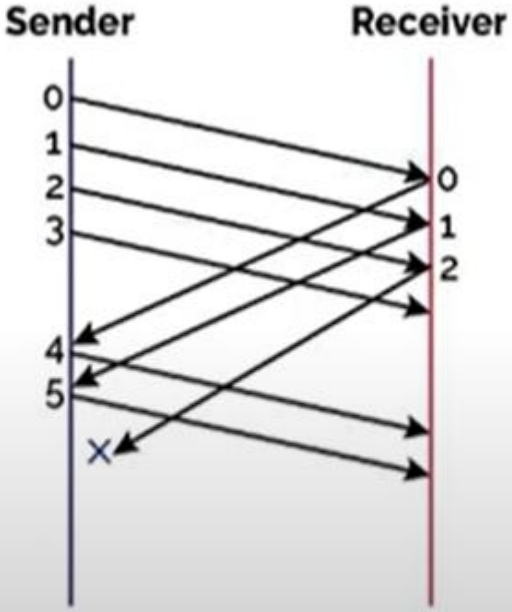
➢ **Step 5:** Sender will then send following frame, which is 5. The window slides containing four frames (2,3,4,5).




Window Size : 4



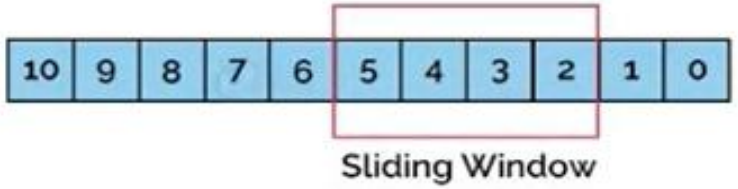

➢ **Step 6:** The acknowledgment for frame no. 2 will then be sent by the receiver. But acknowledgment is lost in network. Not received by the Sender.



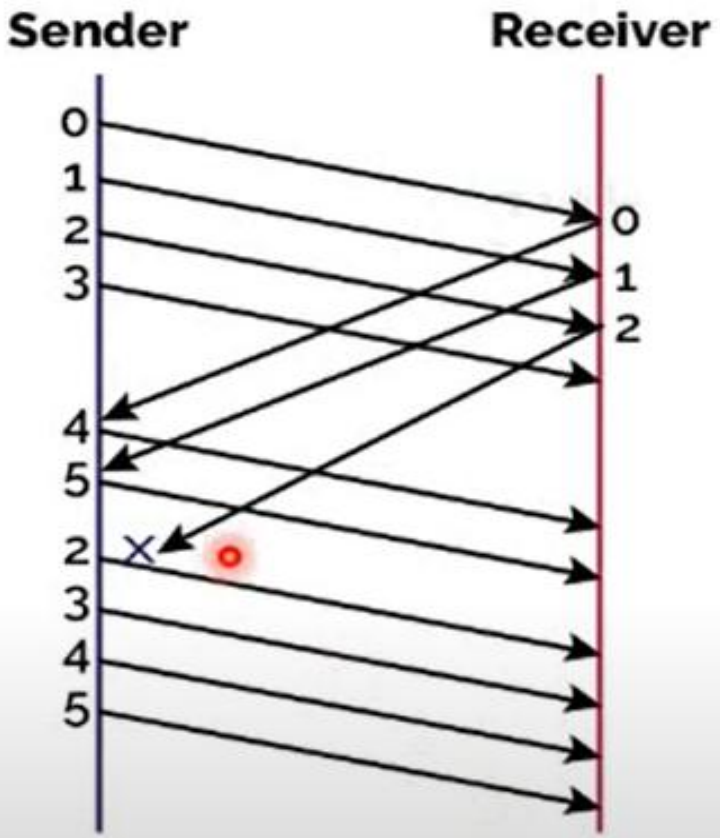
# Go-Back-N ARQ Protocol

➢ **Step 7:** Instead of transmitting frame number 6, the sender Go-Back to 2, which is the initial frame of the current window.

Retransmits all frames in the current window (2,3,4,5).

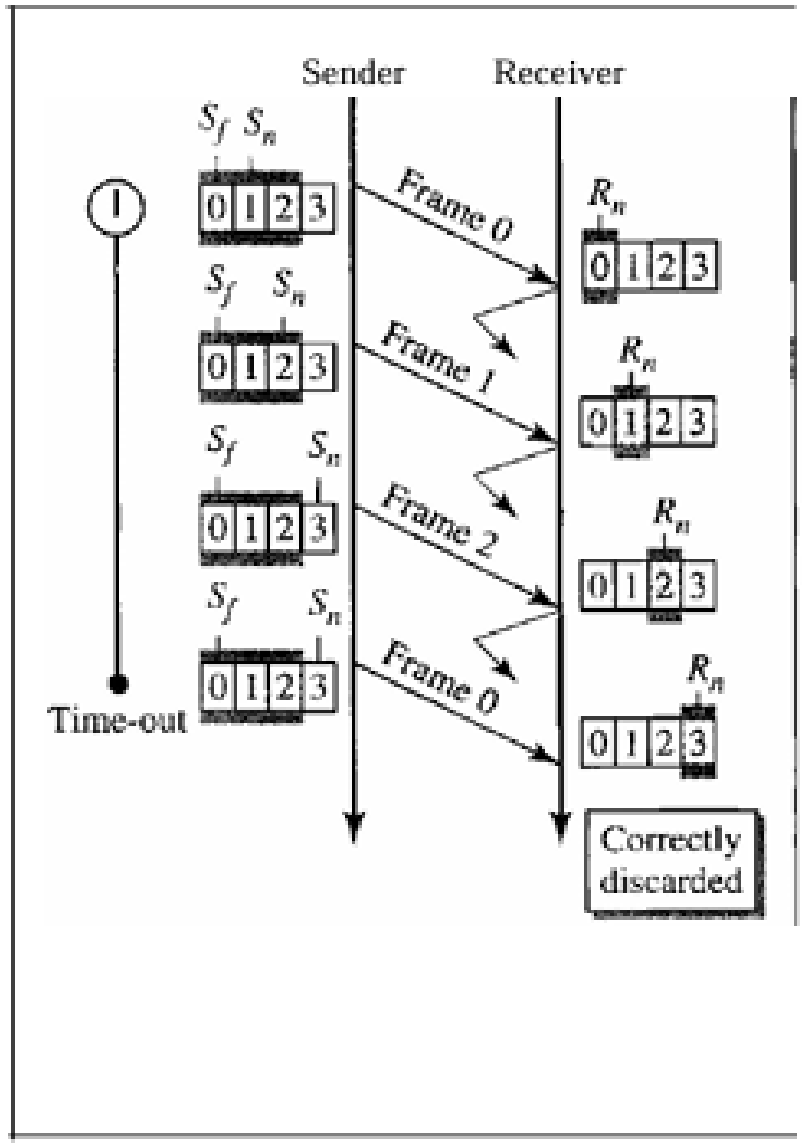



Window Size : 4

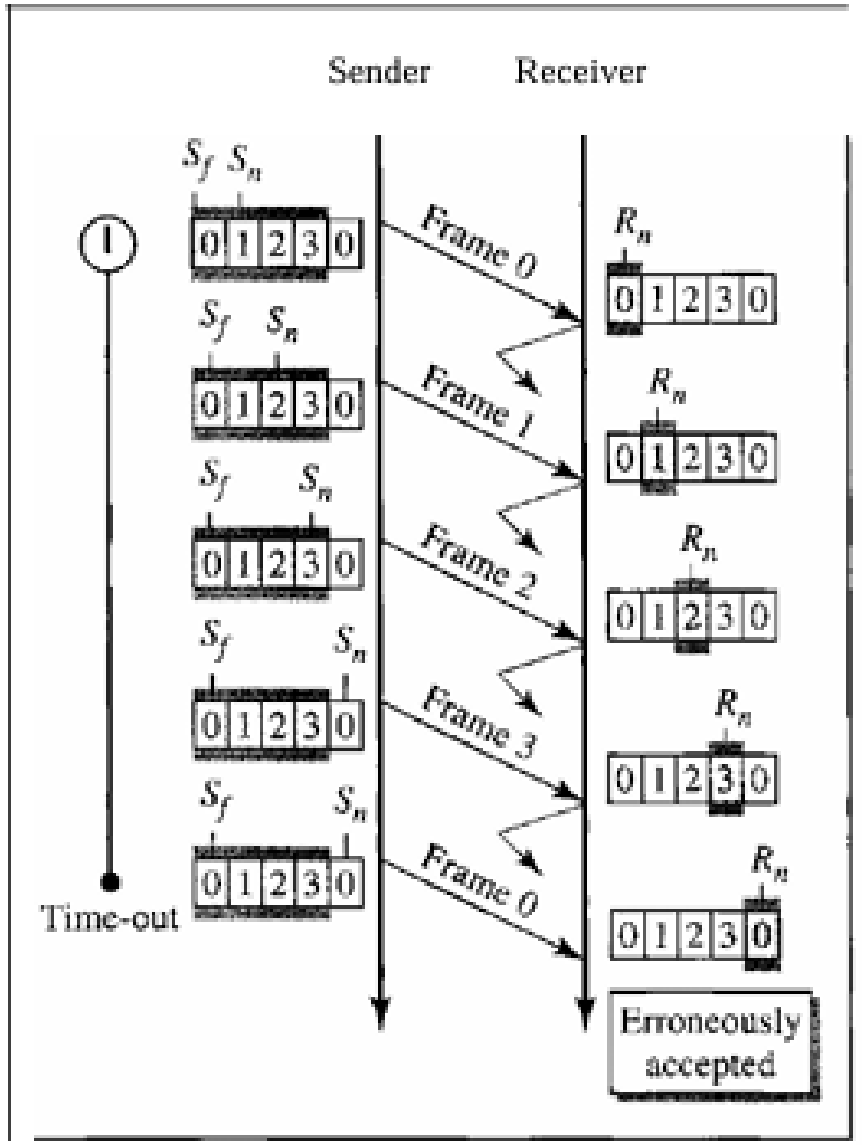


# Go-Back-N ARQ Protocol

- Why the size of the send window must be less than $2^m$ .




a. Window size < $2^m$



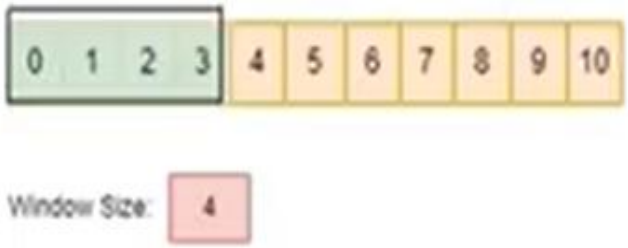

b. Window size = $2^m$

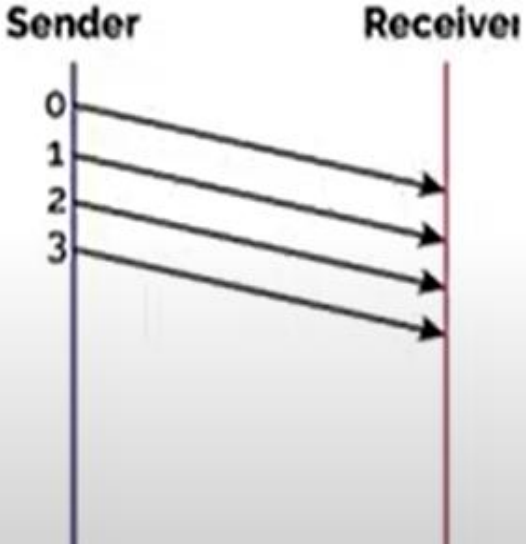
# Selective Repeat ARQ Protocol

- The Go-back-N ARQ protocol works well if it has fewer errors.
- But if there is a lot of error in frame & sending frames again. So, we use Selective Repeat ARQ.
- It stands for Selective Repeat Automatic Repeat Request.
- A sliding window method is used in this data link layer protocol.
- If the receiver receives a corrupt frame, it does not directly discard it.
- It sends a negative acknowledgment to the sender.
- The sender sends that only frame again as soon as on the receiving negative acknowledgment.
- It provide efficient data transmission & fastest recovery.

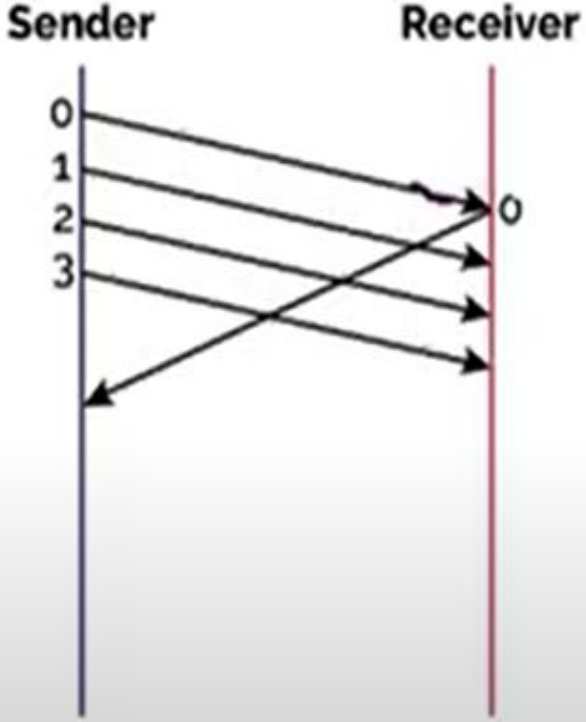
# Selective Repeat ARQ Protocol

- **Step 1:** Sender will send first four frames to the receiver (0,1,2,3). N = 4.



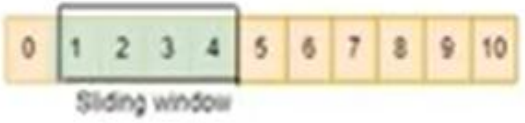




- **Step 2:** Sender expects to get ack from 0th frame.

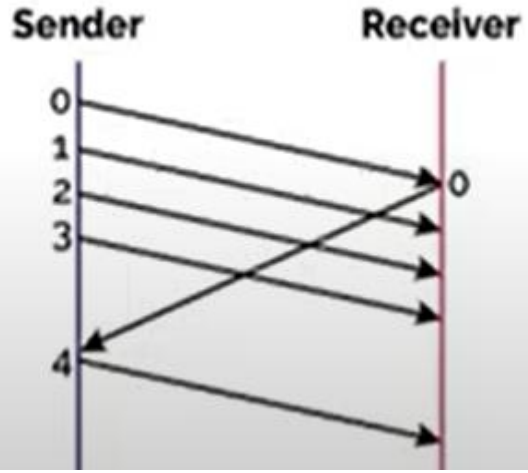


# Selective Repeat ARQ Protocol

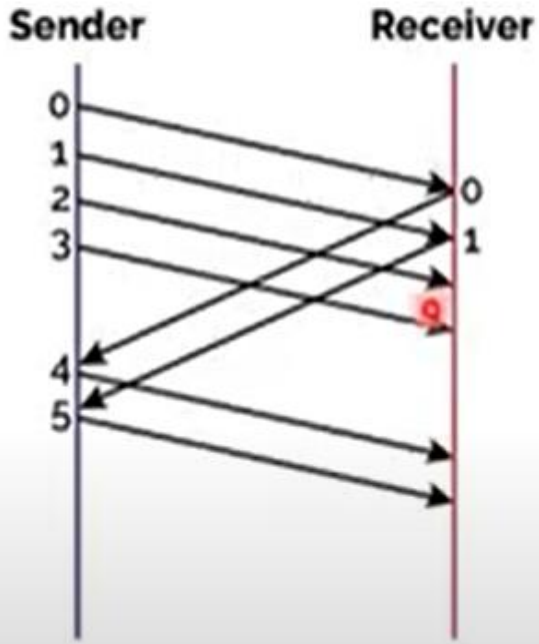
➢ **Step 3:** Sender will send following frame, which is 4. The window slides containing four frames (1,2,3,4).




Window Size: 4



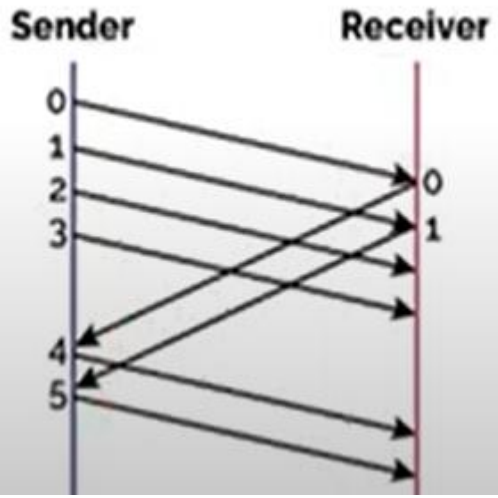

➢ **Step 4:** Sender get ack from 1th frame from the receiver.



# Selective Repeat ARQ Protocol

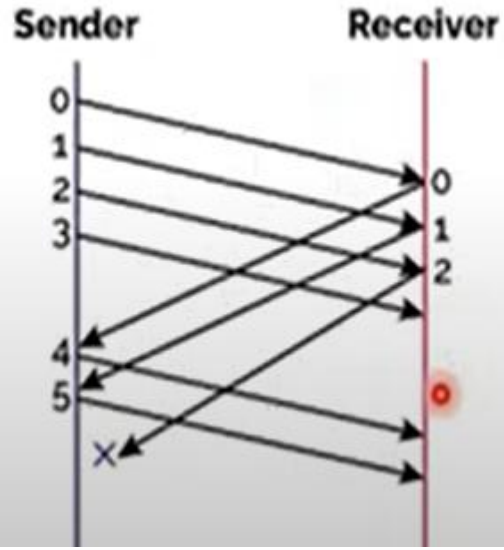
- **Step 5:** Sender will then send following frame, which is 5. The window slides containing four frames (2,3,4,5).


Window Size: 4



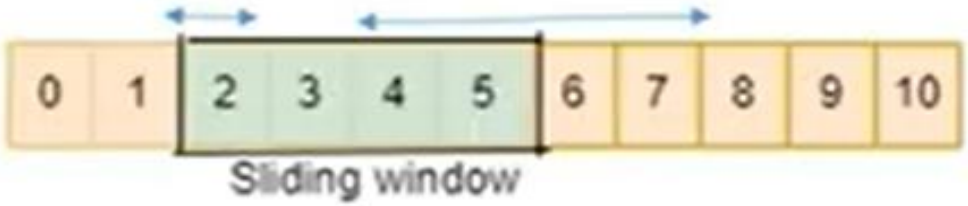

- **Step 6:** The acknowledgment for frame no. 2 will be sent by receiver. But acknowledgment is lost in network.

Then receiver send Negative acknowledgement (NACK) of frame 2 to the sender.



# Selective Repeat ARQ Protocol

➢ **Step 7:** Sender will then send frame 2 alone to the receiver. As usual they send other frames.




Window Size: 4