DLL Protocols

Dr. G. Omprakash

Assistant Professor, ECE, KLEF

Aim of the session

To familiarize students with the basic concept of Elementary Data Link protocols

Learning Outcomes

At the end of this session, you should be able to:

- Summarize protocols for noisy and noiseless channels
- Understand sliding window protocols

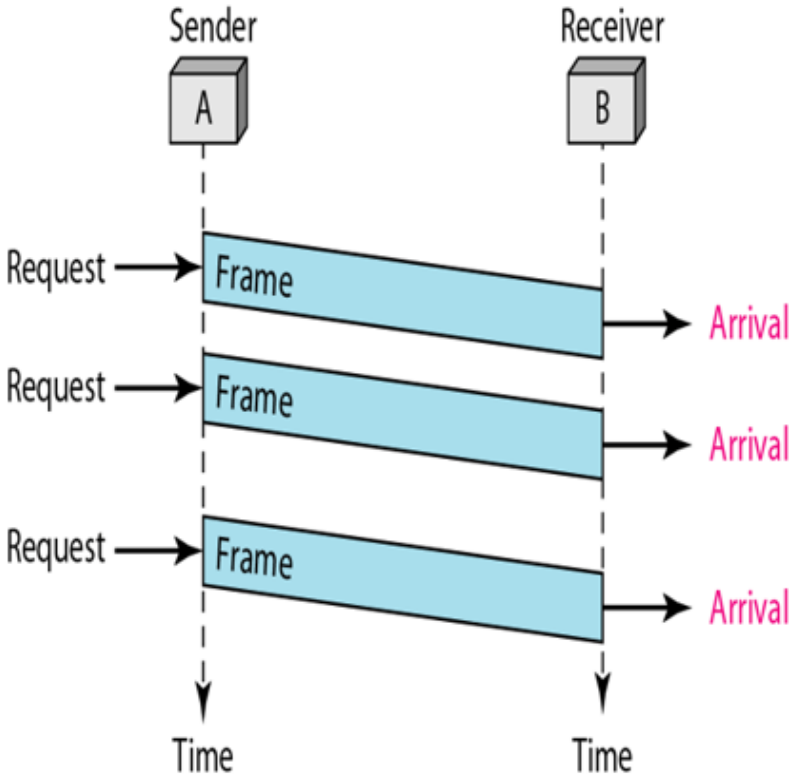
Overview

1. Data link layer protocols
2. Protocols for Noiseless channels
   - Simplest Protocol
   - Stop-and-Wait
3. Protocols for Noisy channels
   - Stop-and-Wait ARQ
   - Go-Back-N ARQ
   - Selective Repeat ARQ

DLL Protocols

- Data link layer protocols deals with **flow control** and **error control**.
- **Flow control** is a technique for assuring that a transmitting entity does not overwhelm a receiving entity with data
- **Error control** in the data link layer is based on automatic repeat request, which is the retransmission of data

- Protocol with no flow or error control.
- The sender sends a sequence of frames without even thinking about the receiver



- Simplest form of flow control
- When data are received, the receiver must do a certain amount of processing
- In the absence of flow control, the receiver's buffer may fill up and overflow
- The source must wait until it receives the acknowledgment before sending the next frame.

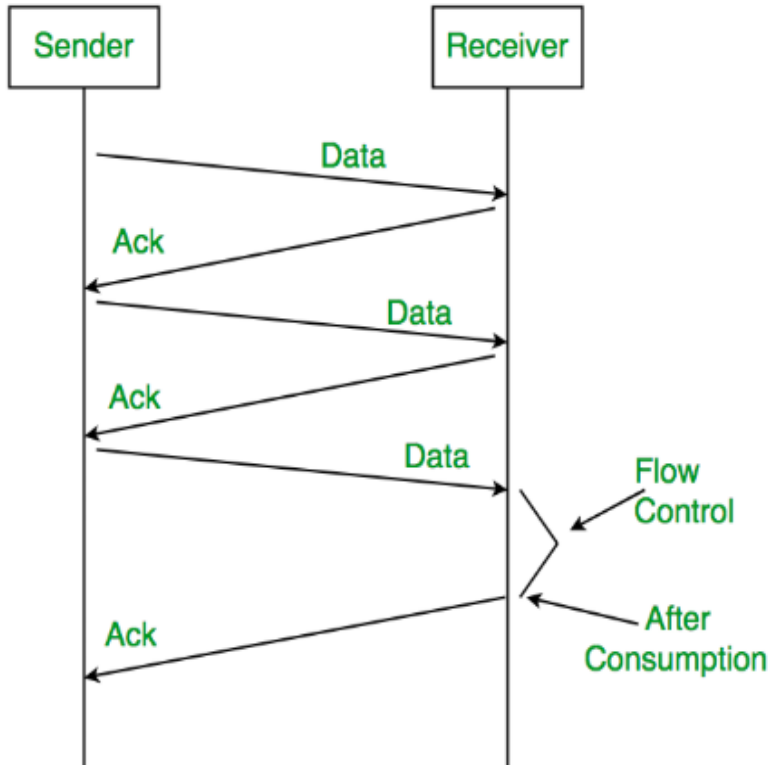


Figure: vertical time sequence diagram

Stop-and-Wait challenges

- **Lost Data** -Receiver waits indefinitely
- **Lost Acknowledgement**-Sender waits indefinitely
- **Delayed Acknowledgement** might be wrongly considered as acknowledgement of some other recent packet

- Frames may be either damaged or lost completely
- if a frame is damaged in transit, the receiver hardware will detect this when it computes the checksum
- If a damaged frame arrived at the receiver, it would be discarded
- If frames are lost, the sender would time out and send the frame again.
- Loss of ACK leads to duplication of frames
- Duplication is avoided using sequence number
- Ambiguity is between two successive frames (0 and 1 bit sequence is sufficient)

- Any arriving frame containing the wrong sequence number is rejected as a duplicate
- In this case, the last valid acknowledgement is repeated so that the sender can eventually discover that the frame has been received
- Protocols in which the sender waits for a **positive acknowledgement** before advancing to the next data item are often called **ARQ (Automatic Repeat reQuest)**

- Stop-and-wait ARQ is a technique used to retransmit the data in case of damaged or lost frames
- Receiver sends the ACK, it includes the number of the next frame that it wants to receive.

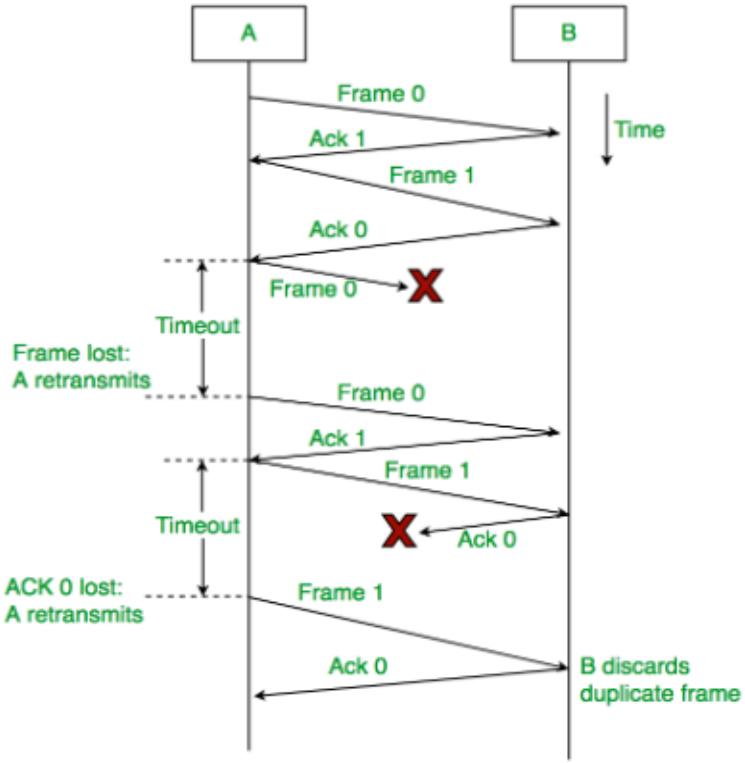
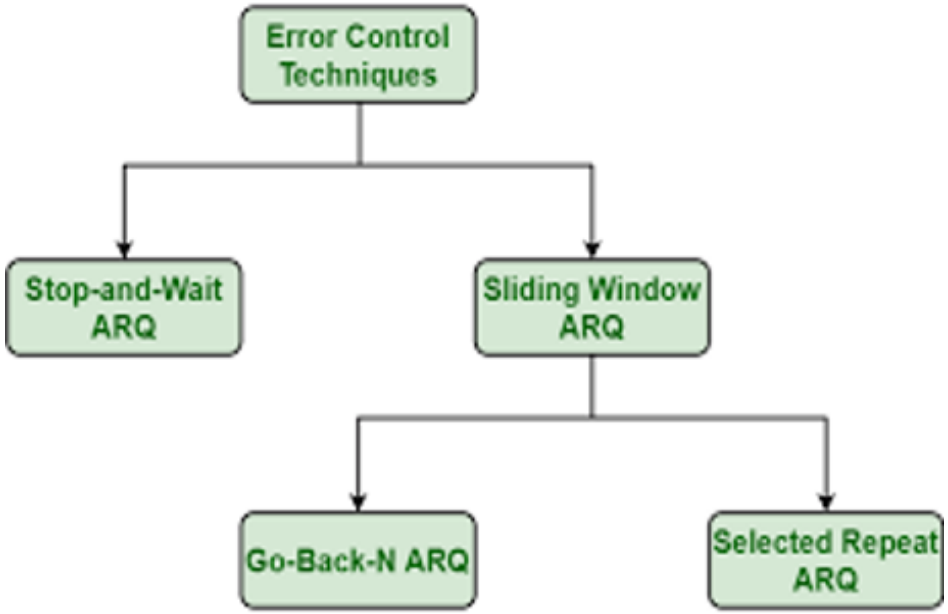
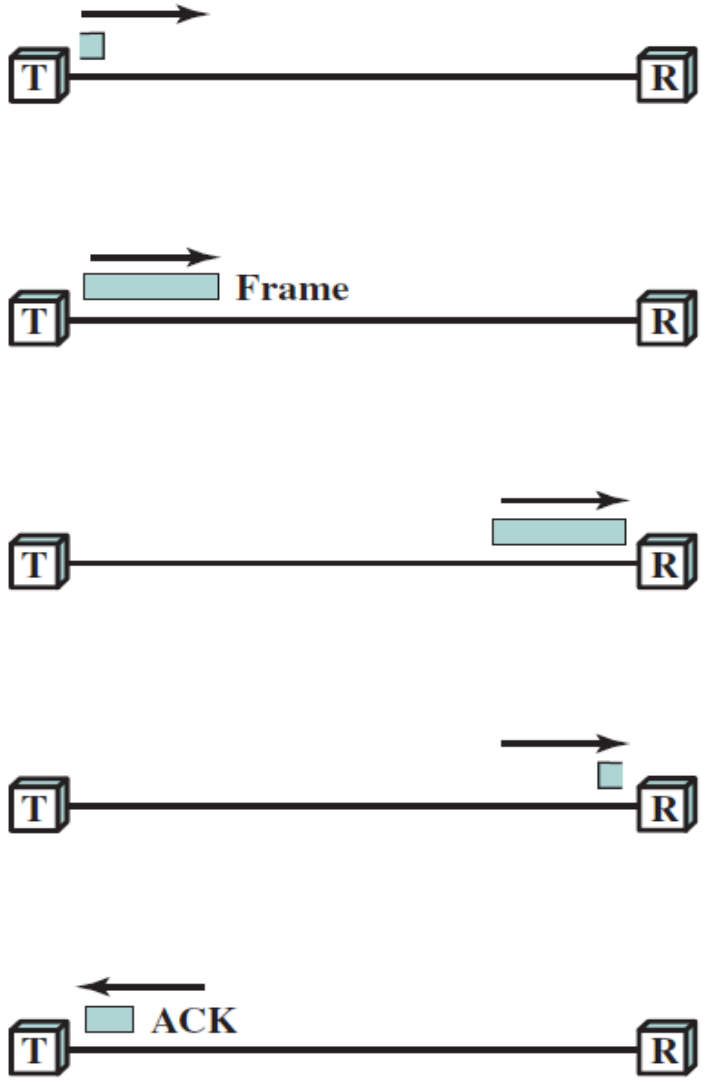


Figure: Vertical time sequence diagram



Motivation for Sliding-Window

- So far- only one frame is transmitted at a time
- If the bit length of the link is much greater than frame length $\Longrightarrow$ Link is always underutilized
- Efficiency can be greatly improved by allowing multiple frames to be in transit at the same time



Sliding Window Protocol

- It is Flow control method in which sender can transmit several frames before getting an acknowledgement
- Capacity of the communication channel can be utilized efficiently.
- A single ACK acknowledge multiple frames.
- When the receiver sends the ACK, it includes the number of the next frame that it wants to receive.
- Sliding Window refers to imaginary boxes at both the sender and receiver end
- It provides the upper limit on the number of frames that can be transmitted before the ACK
- The window has a specific size in which they are numbered as modulo-n (0 to n-1)

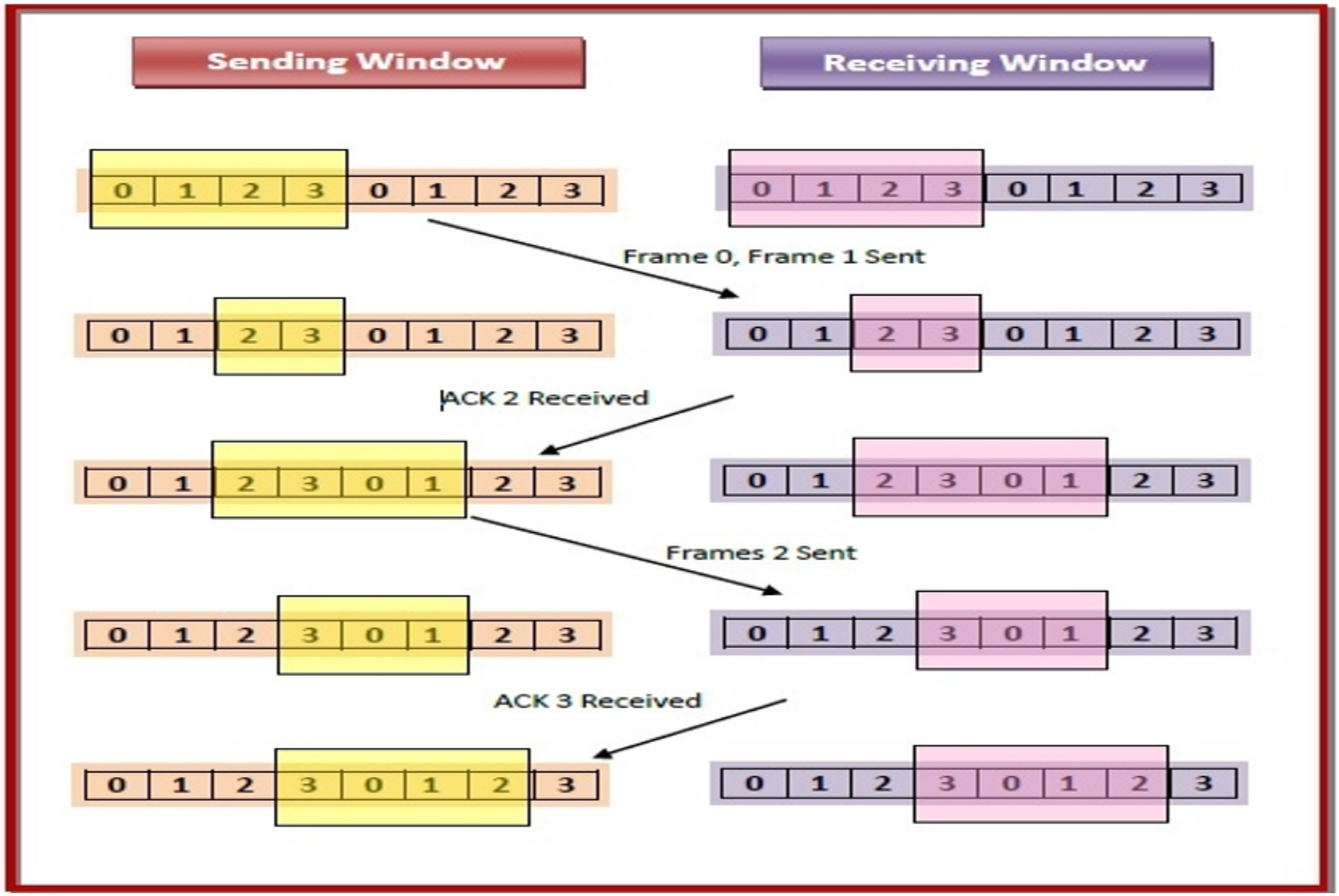


Figure: Go-Back-N, Receiver Window size=4

Go-Back-N ARQ

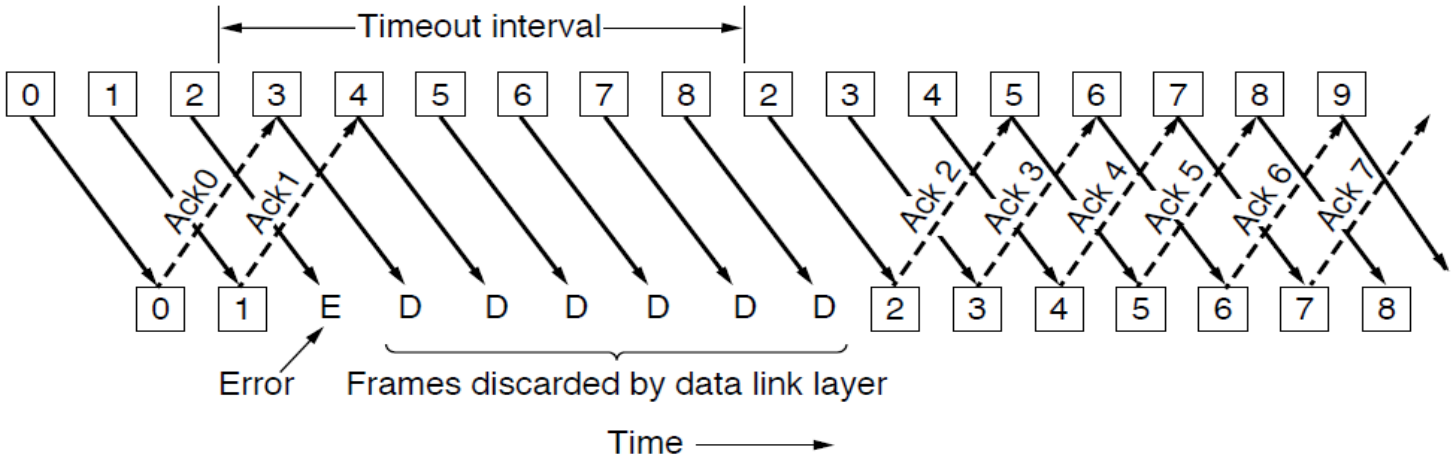


Figure: Go-Back-N, Receiver Window size=1

- Frames 0 and 1 are correctly received and acknowledged
- Frame 2, however, is damaged or lost.
- The sender, unaware of this problem, continues to send frames until the timer for frame 2 expires
- Then it backs up to frame 2 and starts over with it, sending 2, 3, 4, etc. all over again.

- Receiver Window size=1
- If one frame is damaged, receiver just discard all subsequent frames, sending no acknowledgements for the discarded frames
- Data link layer refuses to accept any frame except the next one it must give to the network layer
- Sender will time out and retransmit all unacknowledged frames in order, starting with the damaged or lost one
- This approach can waste a lot of bandwidth if the error rate is high.
- The go-back-n protocol works well if errors are rare

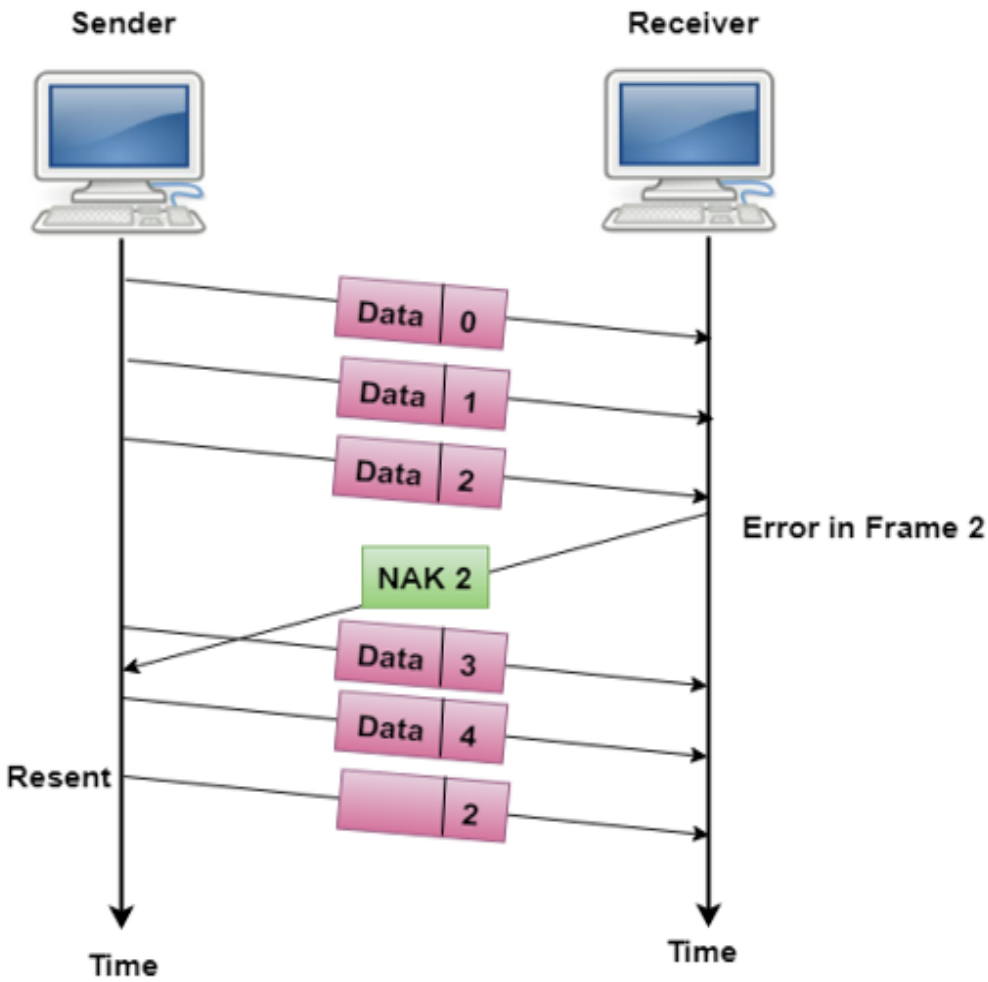


Figure: Selective Repeat, Receiver Window size is large

- Frames 0 and 1–correctly received and acknowledged (frame 2 is lost/error)
- If lost, when frame 3 arrives at the receiver, the data link layer there notices that it has missed a frame
- Receiver sends back a NAK (negative acknowledgement) for frame 2 but buffers frame 3.
- NAK 2 gets back to the sender, which immediately resends frame 2
- If the NAK should get lost, eventually the sender will time out for frame 2 and send it

Selective Repeat ARQ

- Receiver window size is large
- This approach can require large amounts of data link layer memory
- Bad frame that is received is discarded
- Allow the receiver to accept and buffer correct frames received following a damaged or lost one.
- When the sender times out, only the oldest unacknowledged frame is retransmitted

Go-Back-N ARQ	Selective Repeat ARQ
If a frame is corrupted or lost in it, all subsequent frames have to be sent again.	In this, only the frame is sent again, which is corrupted or lost.
If it has a high error-rate, it wastes a lot of bandwidth.	There is a loss of low bandwidth.
It is less complex.	It is more complex because it has to do sorting and searching as well. And it also requires more storage.
It does not require sorting.	In this, sorting is done to get the frames in the correct order.
It does not require searching.	The search operation is performed in it.
It is used more.	It is used less because it is more complex.

Figure: Go-Back-N Vs Selective Repeat

Acknowledge various sources for the images.
Thankyou