Subject: Computer networks

Topic: Error Control

LISNA THOMAS

ACADEMIC YEAR:2020-21

ERROR CONTROL

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.

Error Control

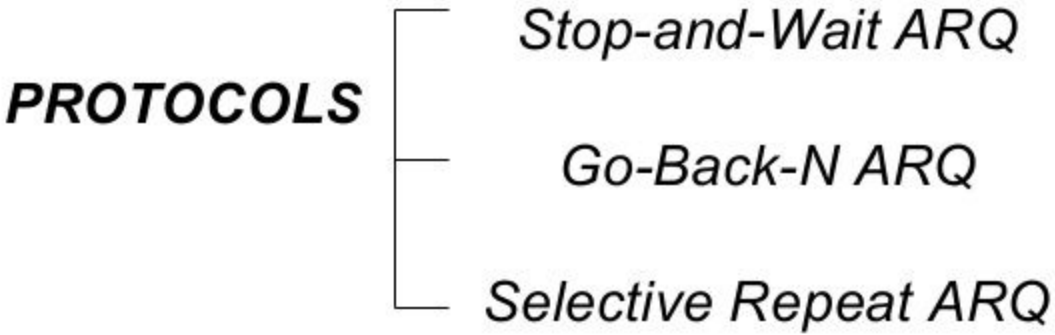
- Having solved the problem of marking the start and end of each frame,
- How make **sure all frames delivered to the network layer destination and proper order.**
- To ensure reliable delivery is to **provide the sender with some feedback about** what is happening
- **If the sender receives a positive acknowledgement about a frame, it knows the frame has arrived safely.**
- **On the other hand, a negative acknowledgement means that something has gone wrong, and the frame must be transmitted again.**

Error Control

- **When the sender transmits a frame, it generally also starts a timer.**
- **The timer is set to expire after an interval long enough for the frame to reach the destination**
- Acknowledgement propagate back to the sender.
- The frame **correctly received and the acknowledgement will get back before the timer runs out**,
- If either the frame or the **acknowledgement is lost**, the timer will **go off,** alerting the sender to a potential problem.
- The obvious solution is to just **transmit the frame again**.
- To prevent this from happening, **it is generally necessary to assign sequence numbers to outgoing frames**, so that the receiver can distinguish retransmissions from originals

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.



Stop-and-Wait ARQ

- 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

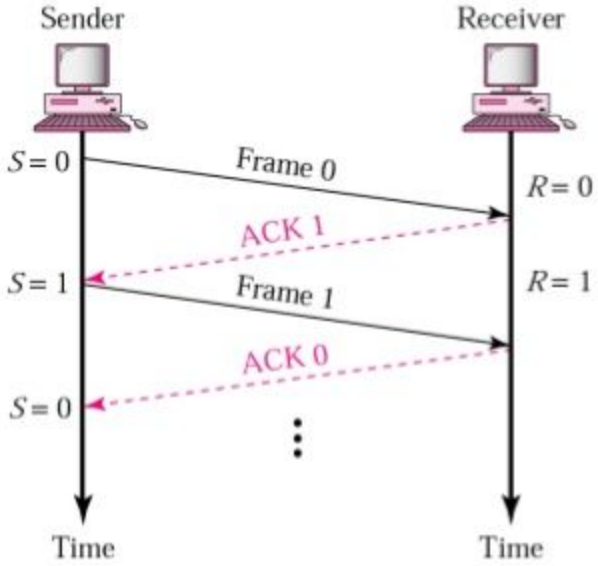
Stop-and-Wait ARQ

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

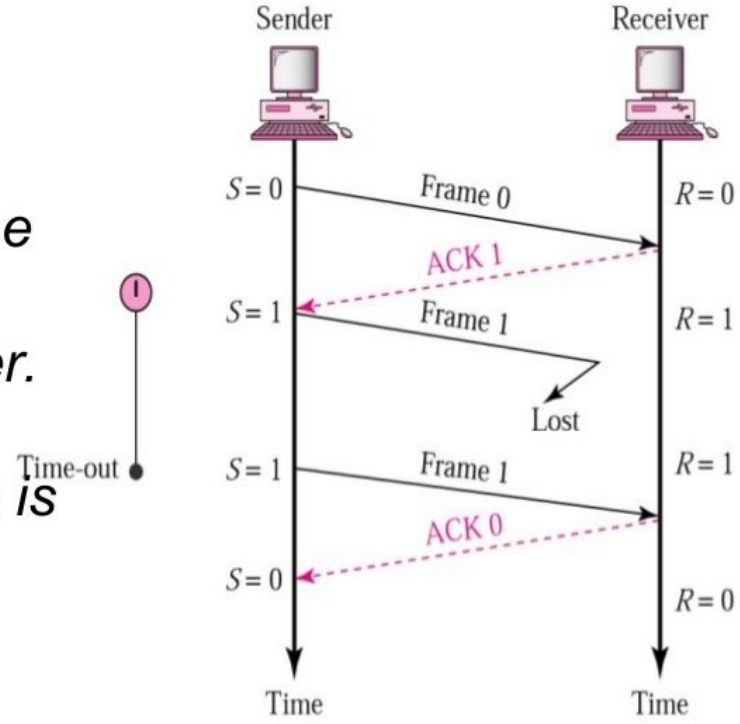
- *The sender will not send the next frame until it is sure that the current one is correctly receive*
- *sequence number is necessary to check for duplicated frames*



1. Stop and Wait ARQ

2. Lost or damaged frame

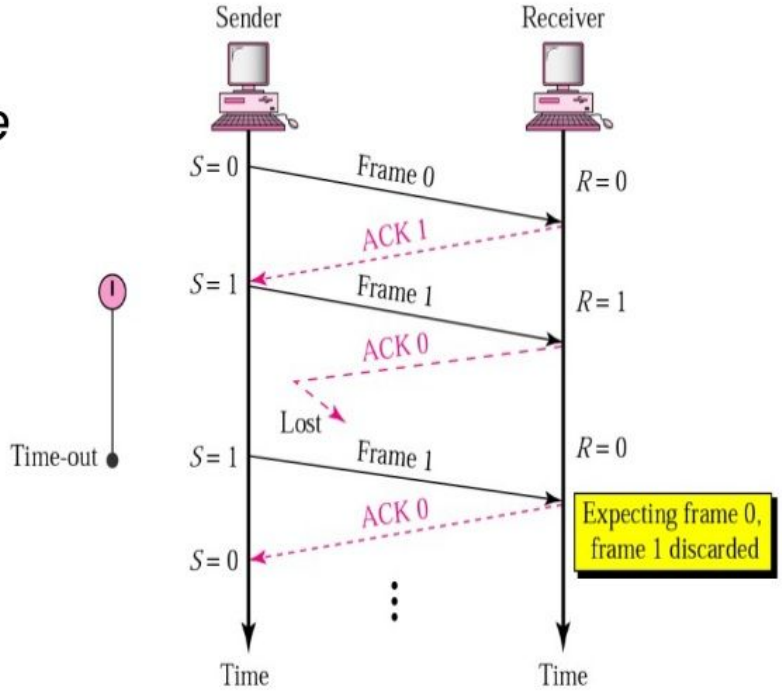
- *A damage or lost frame treated by the same manner by the receiver.*
- *No NACK when frame is corrupted / duplicate*



Stop-and-Wait ARQ

3. Lost ACK frame

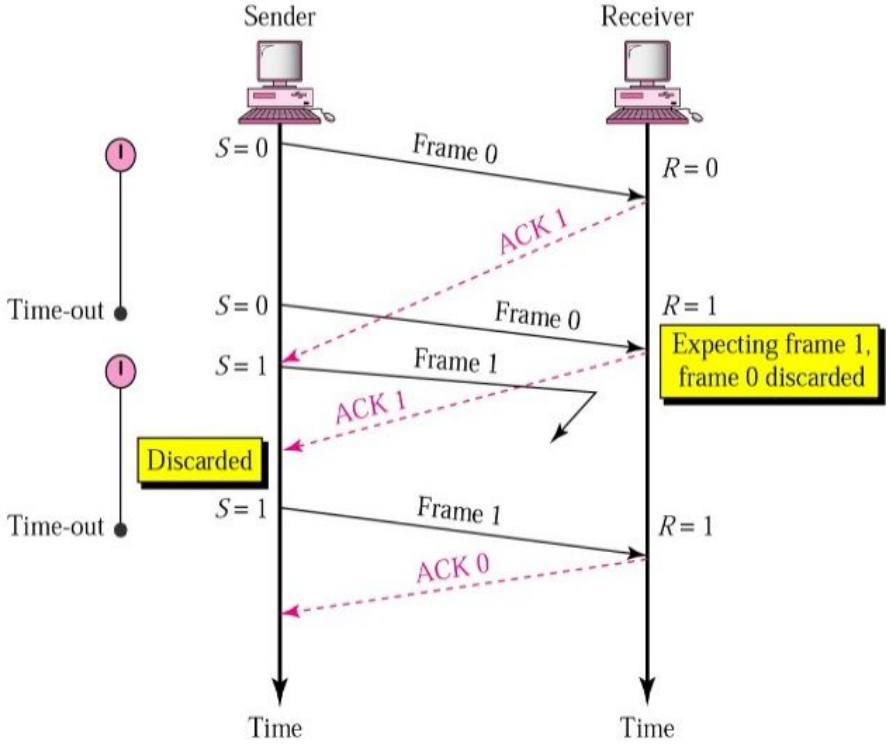
- *Importance of frame numbering*



Stop-and-Wait ARQ

4. Delayed ACK and lost frame

- *Importance of frame numbering*



Go-Back-N ARQ

a) Based on sliding window

b) If no error, ACK as usual with next frame expected

- ACK_i means "I am ready to receive frame i" and "I received all frames between i and my previous ack"

a) Sender uses window to control the number of unacknowledged frames

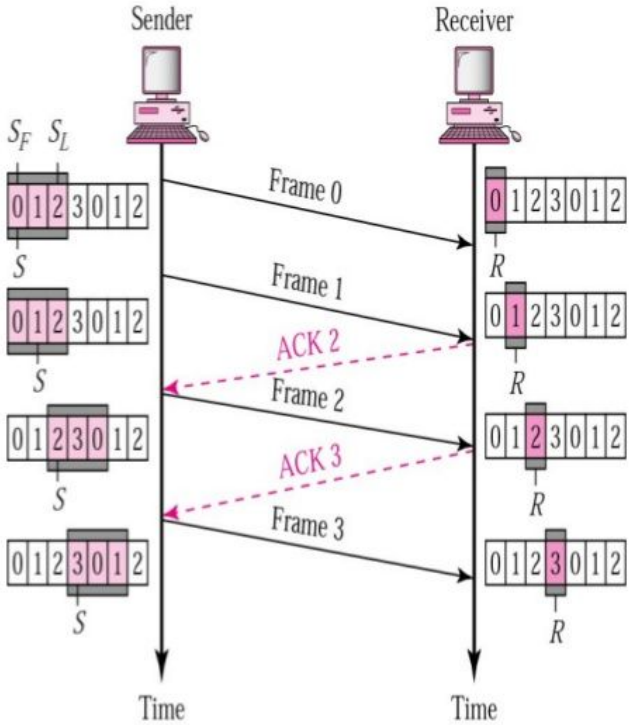
b) If error, reply with rejection (negative ack)

- Discard that frame and all future frames until the frame in error received correctly
- Transmitter must go back and retransmit that frame and all subsequent frames

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 - Damaged Frame

a) Receiver detects error in frame *i*

b) Receiver sends "reject *i*"

c) Transmitter gets "reject *i*"

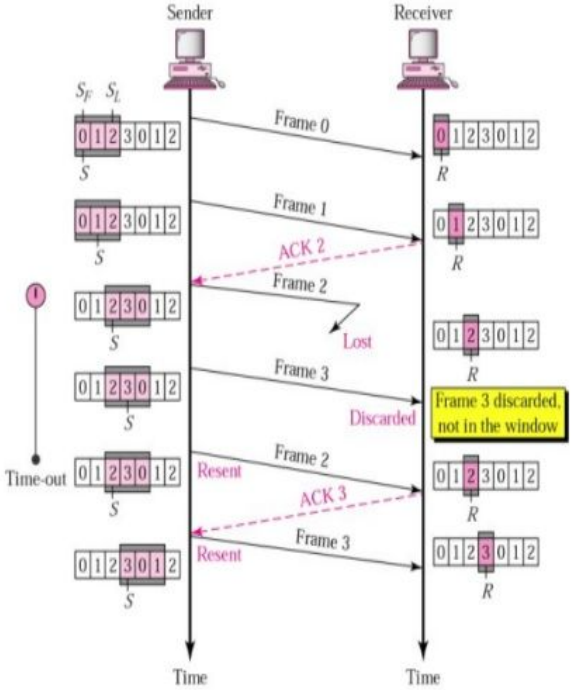
d) Transmitter retransmits frame *i* and all subsequent frames

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 - Lost Frame

a) Frame *i* lost

b) Transmitter sends frame *i*+1

c) Receiver gets frame *i*+1 out of sequence

d) Receiver sends "reject *i*"

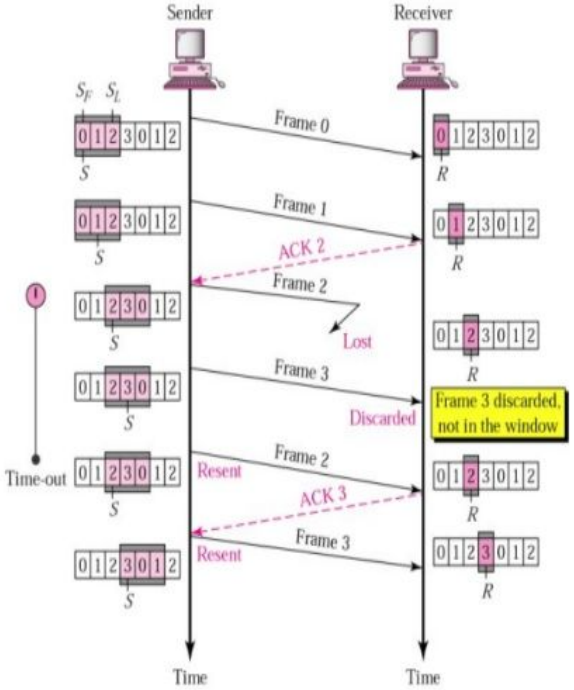
e) Transmitter goes back to frame *i* and retransmits it and all subsequent frames

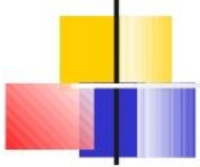
Go-Back-N ARQ

Damage or Lost Frame

Correctly received out of order packets are not Buffered

What is the disadvantage of this?





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