3.4 Flow Control and Error Control

Error Control

- Network is responsible for transmission of data from one device to another device.
- The end to end transfer of data from a transmitting application to a receiving application involves many steps, each subject to error.
- With the error control process, we can be confident that the transmitted and received data are identical.
- Data can be corrupted during transmission.
- For reliable communication, error must be detected and corrected.

Flow Control

- Flow Control is one important design issue for the Data Link Layer that controls the flow of data between sender and receiver.
- Error control and flow control in data link layer is done by using Automatic Repeat Request (ARQ) mechanism.

ARQ:

- Once an error is detected in an transmission (data lost, damaged frames, lost acknowledgment), the specified frames are retransmitted. This process is called automatic repeat request (ARQ).

Two types of ARQ

A. Stop-and-Wait Protocol

B. Sliding Window Protocol

1. Go Back N ARQ
2. Selective Repeat ARQ

3.4.1 Stop-and-Wait Protocol

The sender sends a frame, stops sending next frame until it receives acknowledgment for the previous frame.

Step 1:

- The data frames are numbered alternatively as 0 and 1's.
- The data frame 0 is acknowledged by acknowledgment (Ack) 1 (requesting sender to send the next frame with no.1)

Step 2 :

- The sender has a control variable "S" that holds the number of recently sent frame.
- The receiver has a control variable "R" that holds the number of next frame expected by the receiver (previous frames are successfully received and acknowledged . So it expects the next frame).

Step 3 :

- The sending device holds a copy of the frame transmitted.
- If the ack is received, the copy is deleted.
- Otherwise the copy is used for retransmission.

Step 4:

- The sender starts a timer when it sends a frame. If the acknowledgment is not received in the specified time, the frame is retransmitted.

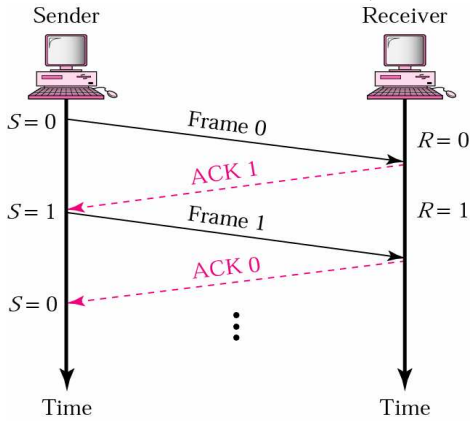
- If the ack is received within specified time, the next frame is sent and its timer is started.

Step 5:

- The receiver sends an ack for the frames that are received correctly.
- If the frames received, are damaged or the frames are lost in transmission then the receiver does not sends any acknowledgment. So when the timer expires, the sender understands that the receiver does not receive the frame and it retransmits the frame.

1. Normal Operation :

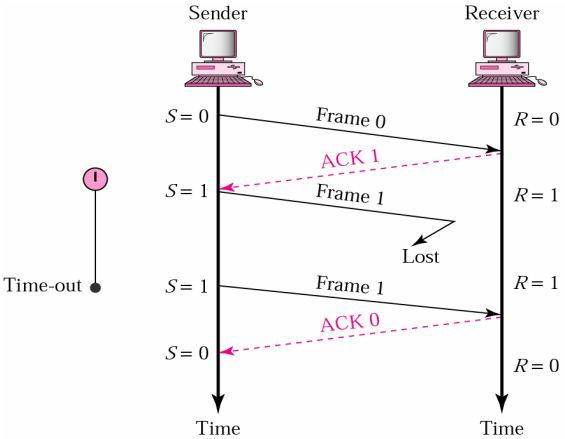
- The frame 0 is sent successfully (without damage or loss of frame)
- The Ack 1 is sent back by the receiver to the sender.
- The sender sends the next frame, frame1 and waits for ack 0 and so on.



- All the frame are sent successfully and ack are received before the timer expires.

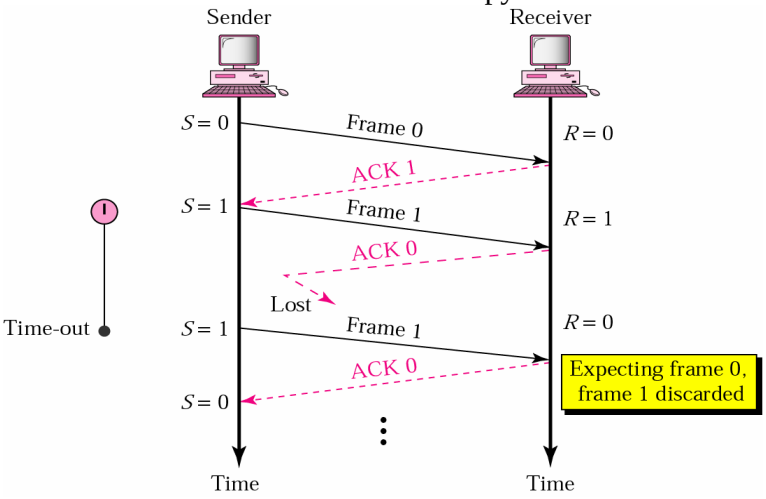
2. Lots or damaged frame.

- If the frame sent, is lost in the transmission, then the ack is not sent for that frame.
- If the frame is received with error then the frame is discarded and ack is not sent for that frame when the timer expires the sender retransmits the lost / damaged frame.



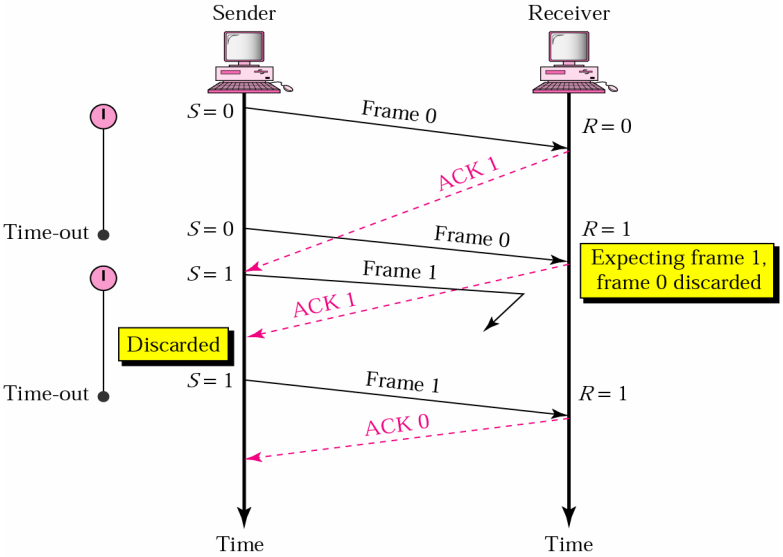
3. Lost Acknowledgment :

- If the ack is lost, then the sender waits for the specified time and assumes that the frame is not received by the receiver successfully. So it retransmits the data frame.
- The receiver has already received the frame which is retransmitted. So the receiver discards the second copy of the frame and it sends the ack for that frame.



3.Delayed Acknowledgment :

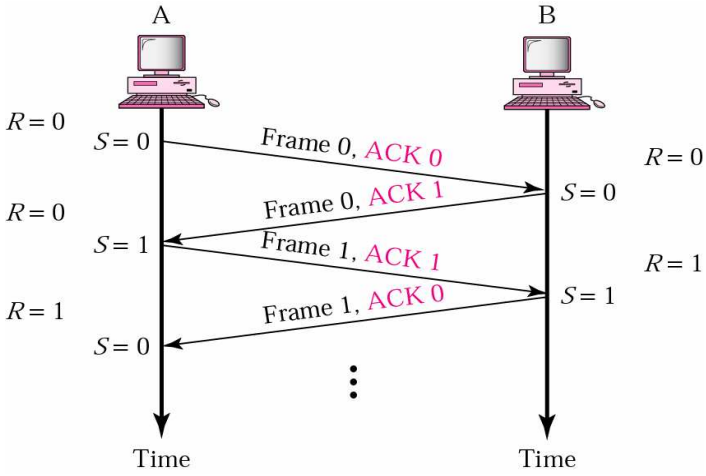
- If an ack does not reaches the sender within the specified time, the sender assumes that
- the frame sent may be lost and so it retransmits the frame.
- The receiver has successfully received the frame and the second copy of the frame retransmitted is discarded.
- The ack which reaches after the specified time is accepted and the next frame is sent.
- The ack for the second copy is discarded in the sender side.



Piggybacking :

- During each transmission the frame is sent along with the ack for the previous frame.

- This method of combining data frame with an ack is called piggybacking.
- Piggybacking saves bandwidth.
- The overhead like addresses, CRC computation is done only once for both frame ack, instead of computing it separately for frame and ack.



Disadvantage of stop and wait :

- The sender waits for more time till it receives the acknowledgment for each frame.
- The transmission medium is not efficiently utilized.
- The time taken for sending "n" frame is more.

3.4.2 Sliding Window Protocol

- Each frame is assigned a unique consecutive sequence number.
- The receiver uses the numbers to place received frames in the correct order, discarding duplicate frames and identifying missing ones.

Sender

The sender assigns a sequence number, denoted **SeqNum**, to each frame.
Assume that it can grow infinitely large.
The sender maintains three variables:

1. The send window size, denoted **SWS**, gives the upper bound on the number of outstanding (unacknowledged) frames that the sender can transmit;
2. **LAR** denotes the sequence number of the last acknowledgment received;
3. **LFS** denotes the sequence number of the last frame sent. The sender also maintains the following invariant:

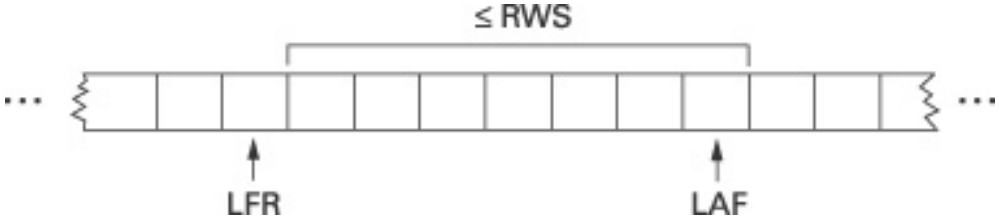
$$LFS - LAR \leq SWS$$

- When an acknowledgment arrives, the sender moves LAR to the right, thereby allowing the sender to transmit another frame. ie slide the window right.
- The sender associates a timer with each frame it transmits, and it retransmits the frame when the timer expires before an ACK is received.
- The sender has to buffer up to SWS frames, since it must be prepared to retransmit them until they are acknowledged.

The receiver maintains the following three variables:

1. The receive window size, denoted **RWS**, gives the upper bound on the number of out-of-order frames that the receiver is willing to accept.
2. **LAF** denotes the sequence number of the largest acceptable frame
3. **LFR** denotes the sequence number of the last frame received. The receiver also maintains the following invariant:

$$LAF - LFR \leq RWS$$



When a frame with sequence number SeqNum arrives, the receiver takes the following action.

- If SeqNum ≤ LFR or SeqNum > LAF, then the frame is outside the receiver's window and it is discarded.
- If the SeqNum is within the window and arrived in out of order(Expected Frame lost or damaged) buffer the received frame and send a NAK to the sender for the expected frame.
- If LFR < SeqNum ≤ LAF, then the frame is within the receiver's window and it is accepted. If the SeqNum received in order slide the window one position right.
- Now the receiver needs to decide whether or not to send an ACK.
- Let SeqNumToAck denote the largest sequence number not yet acknowledged, such that all frames with sequence numbers less than or equal to SeqNumToAck have been received.
- The receiver acknowledges the receipt of SeqNumToAck, even if higher-numbered packets have been received
- This acknowledgment is said to be cumulative.
- It then sets LFR = SeqNumToAck and adjusts LAF = LFR + RWS.

Example

- Consider LFR = 5 (i.e., the last ACK the receiver sent was for sequence number 5 ie ACK5), and RWS = 4. This implies that LAF = 9. Should frames 7 and 8 arrive, they will be buffered because they are within the receiver's window.
- However, no ACK needs to be sent since frame 6 is yet to arrive.
- Frames 7 and 8 are said to have arrived out of order.
- Send a NAK6 to the sender saying receiver is expecting Frame6.
- On receiving NAK6 sender understands Frame6 is lost or damaged and so the sender retransmits Frame6.
- When the receiver receives Frame6, it acknowledges frame 8, bumps LFR to 8, and sets LAF to 12.

Finite Sequence Numbers and Sliding Window

If the header allocates m bits for sequence number (SeqNum), then the range of sequence number is from 0 to 2^m-1 .

Example: If $m = 2$

$0 – (2^2–1)$

0 – 3 is the range. ie., 0, 1, 2, 3

- The number of available sequence numbers denoted by **MaxSeqNum** is 2^m
- If there are 9 frames to be sent, and 2 bits (m=2) are allotted in the header of frame for SeqNum, the frames are numbered as : 0 1 2 3 0 1 2 3
- The sequence is repeated
- SWS ≤ MaxSeqNum − 1

Go Banck N ARQ

Sender Sliding window:

- The frames are stored in a buffer.
- The frames that are sent but the ack is expected (outstanding frames) are enclosed in a window.
- The size of the window is $2^m–1$
- On receiving an ack the window slides.

Receiver sliding Window :

- The window size = 1 in the receiver side.
- The receiver is expecting a specific frame to arrive in specific order

Timers :

- Sender starts a timer for every frame that was sent.
- Receiver has not timers.

Acknowledgment :

- If the frame is received without damage then the receiver sends an positive acknowledgment, requesting the next fame.
- If the frame received is damaged or out of order other than expected frame then the receiver keeps silent without sending acknowledgment. So the timer in sender side expires and the sender retransmits the frames starting from the one with expired timer.
- Can send single or cumulative ack.
- No concept of NAK

Example :

0 1 2 3 are the frames

. If the timer for frame 1 has expired, but frame 2 and 3 are already sent. So from frame 1 all the frames are retransmitted.

Retransmitted frames : 1,2, 3.

The sender go back and send N frames. So the protocol is called Go Back N ARQ.

Normal Operation :

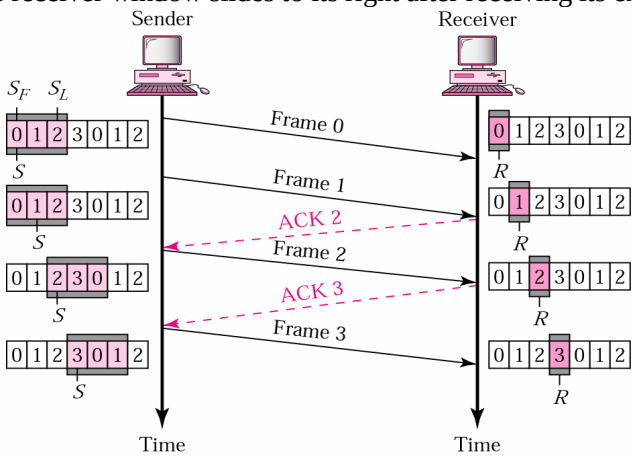
For *n* frames inside Window only one ack can be sent. (For 0 & l, only one ack, ack 2 is sent.)

Sender :

• The frames in the window are sent sequentially without waiting for the ack.
• The ack can be accepted as cumulative ack. That is for all the frames in the window one ack is received. Ex : for 0,1,2 frames, after the receiver receives 0,1,2 it send only one ack, ack 3 which
means all the previous frames are received successfully.
• Once the ack is got the sender window slides to its right and the control variable are updated.

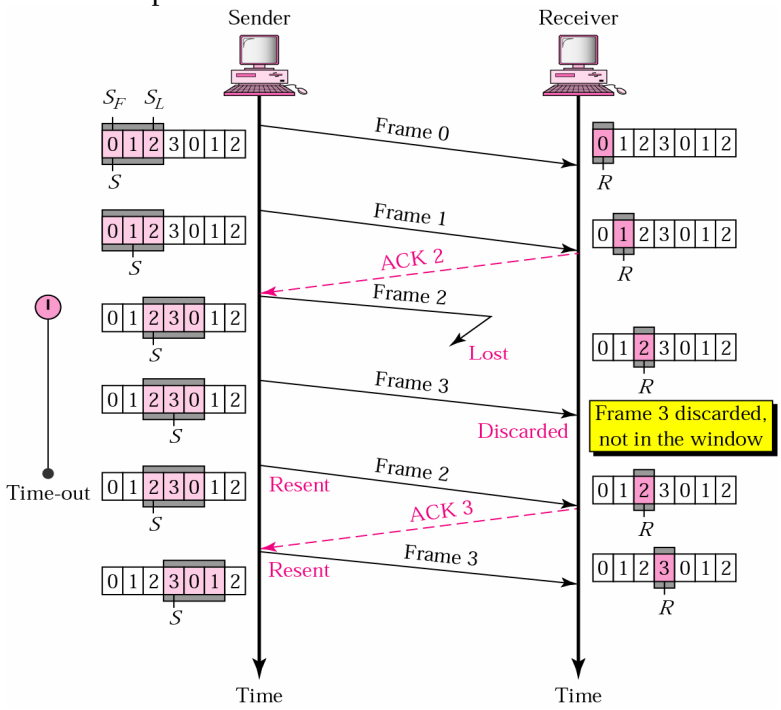
Receiver :

The receiver window slides to its right after receiving its expected frame.



Damaged or lost Frame:

If any frame is lost or damaged the sender sends the next frame sequentially. The receiver expecting a specified frame (lost frame) discards all other incoming frame. The sender timer expires and it retransmits from the lost frame.



Lost acknowledgment :

The frame is resend if the ack doesn't reach the sender at time or lost or damaged if the

previous ack are lost but the cumulative ack reaches the sender, then there is no need for retransmission.

Delayed Acknowledgment :

If the ack doesn't reach the sender in specified time then the frame is resend.

Slective Repeat ARQ

In goback N if one frame is damaged or lost N frames are resend.
To avoid this overhead, the selective repeat ARQ uses a mechanism to retransmit only the damaged or lost frame instead of N frames.

Sender Sliding window:

- The frames are stored in a buffer.
- The frames that are sent but the ack is expected (outstanding frames) are enclosed in a window.
- The size of the window is $2^m/2$
- On receiving an ack the window slides.

Receiver Window :

- Receiver Window size is same as the sender window size.
- After a frame is received successfully they receiver sends an ack to the sender.
- The receiver window sides to its right side if the frame is received successfully.

Normal Operation :

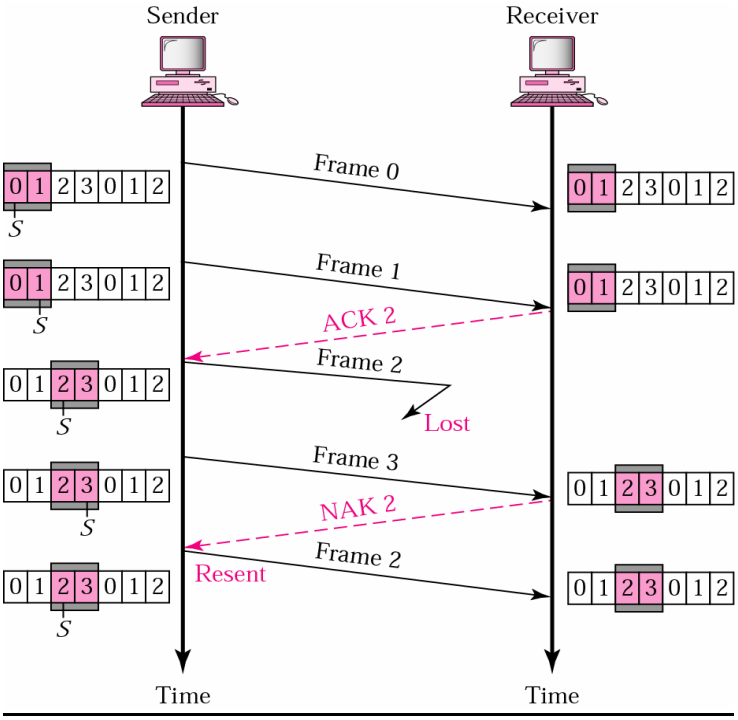
Sender sends the frames in the window one by one. After the ack is received the sender window slides to its right.

Frame lost or damage:

- If frame 2 is lost, frame 3 is sent by the sender sequentially.
- The Receiver accepts frame 3 if frame 3 is in receiver window and sends and NAK 2 to sender requesting for the retransmission of frame 2.

Negative Acknowledgment (NAK):

- If a frame is damaged or lost the receiver sends on NAK to the sender so that the sender retransmits the lost or damaged frame.



Lost and delayed Ack's:

- The sender sets a timer when a frame is sent. If NAK or Ack is not received in the specified time the sender retransmits the frame.