Data Link Control Layer (DLC)

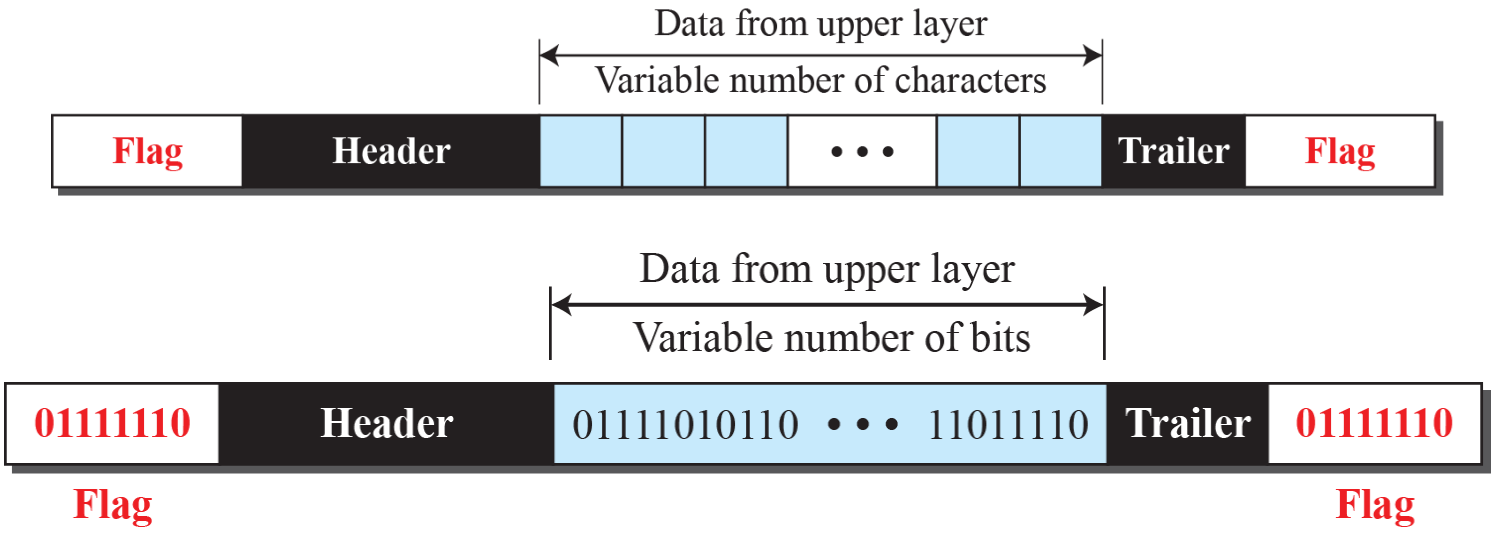
Session Objectives

After going through this session you will be able to understand

- ✓ Flow control
- ✓ Protocols for flow control

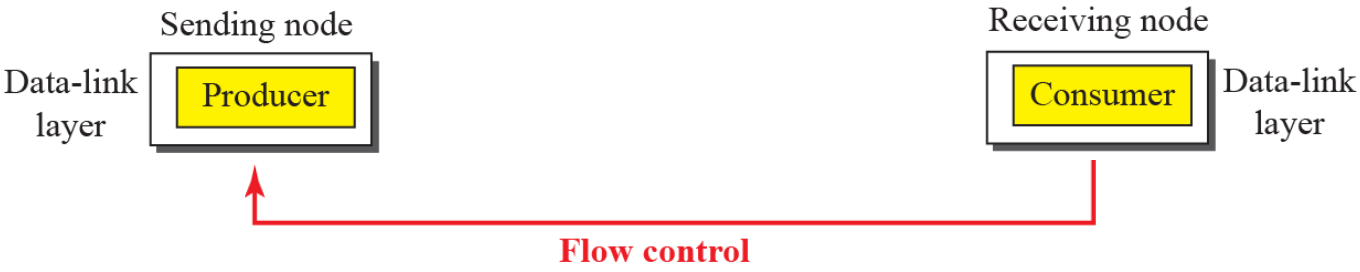
DLC Services

- The data link control (DLC) deals with procedures for communication between two adjacent nodes no matter whether the link is dedicated or broadcast.
- Data link control functions include framing, flow and error control.
- Our postal system practices a type of framing. The simple act of inserting a letter into an envelope separates one piece of information from another; the envelope serves as the delimiter.
- Framing in the data-link layer separates a message from one source to a destination by adding a sender address and a destination address. The destination address defines where the packet is to go; the sender address helps the recipient acknowledge the receipt.



Flow and Error Control

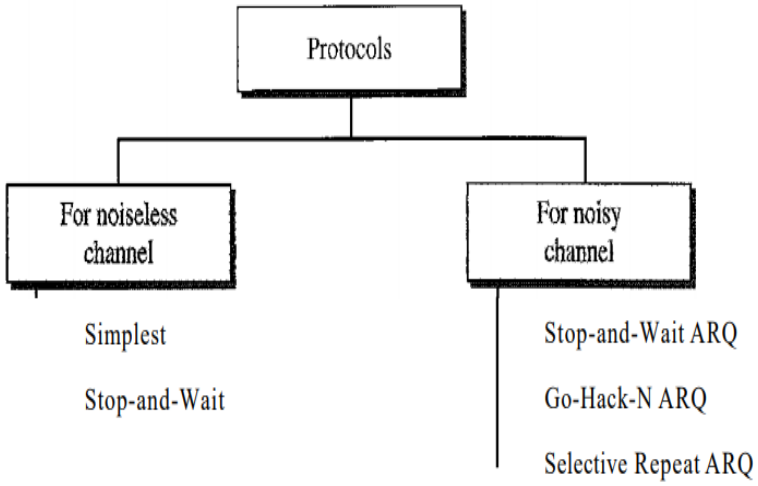
- One of the responsibilities of the data-link control sublayer is flow and error control at the data-link layer.
- Traditionally four protocols have been defined for the data-link layer to deal with flow and error control: Simple, Stop-and-Wait, Go-Back-N, and Selective-Repeat.



Protocols:

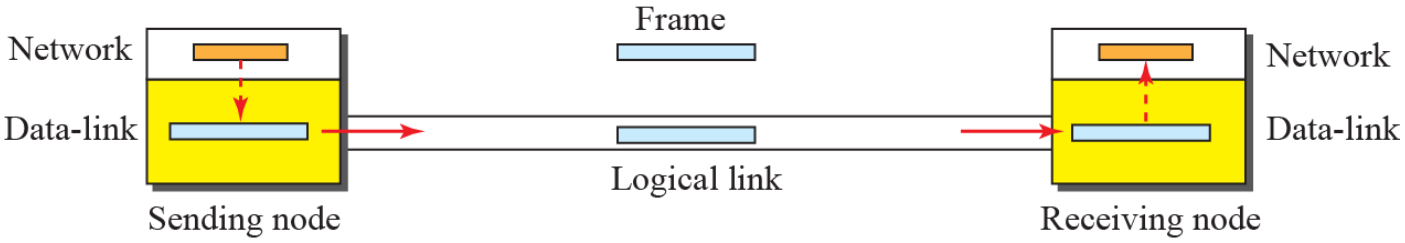
- How the data link layer can combine framing, flow control, and error control to achieve the delivery of data from one node to another. The protocols are normally implemented in software by using one of the common programming languages

- We divide the discussion of protocols into those that can be used for noiseless (error-free) channels and those that can be used for noisy (error-creating) channels. The protocols in the first category cannot be used in real life, but they serve as a basis for understanding the protocols of noisy channels. Figure shows the classifications.

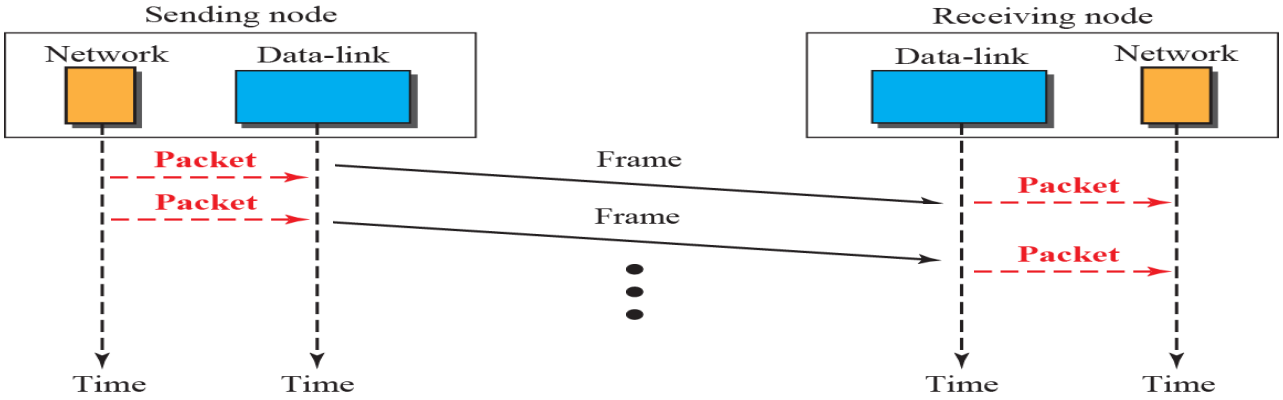


NOISELESS CHANNELS
Simple Protocol

- Simple protocol has neither flow nor error control.
- It assumes that the receiver can immediately handle any frame it receives. In other words, the receiver can never be overwhelmed with incoming frames.



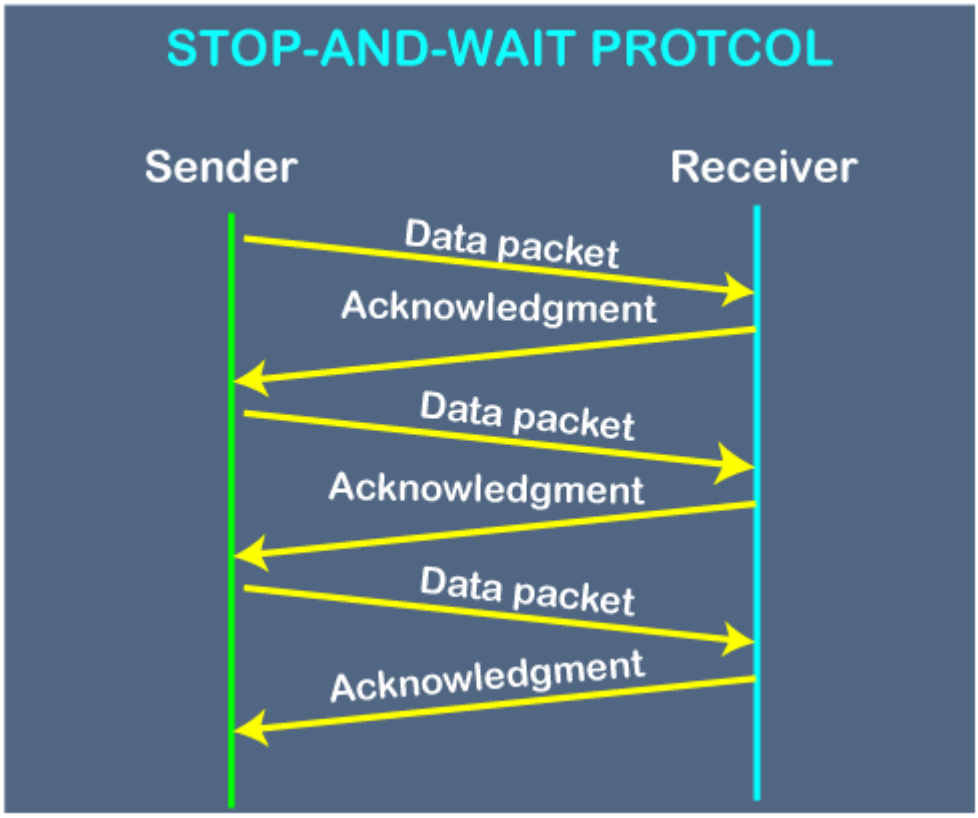
Example(Shown Below): The sender sends frames one after another without even thinking about the receiver.



Stop-and-Wait Protocol

- If data frames arrive at the receiver site faster than they can be processed, the frames must be stored until their use.
- Normally, the receiver does not have enough storage space, especially if it is receiving data from many sources. This may result in either the discarding of frames or denial of service.
- To prevent the receiver from becoming over- whelmed with frames,we somehow need to tell the sender to slow down. There must be feedback from the receiver to the sender.
- The protocol we discuss now is called the Stop-and-Wait Protocol because the sender sends one frame, stops until it receives confirmation from the receiver (okay to go ahead), and then sends the next frame.
- We still have unidirectional communication for data frames, but auxiliary ACK frames (simple tokens of acknowledgment) travel from the other direction.

Stop-and-Wait Protocol



Automatic Repeat ReQuest (ARQ)

- **It** is a group of error – control protocols for transmission of data over noisy or unreliable communication network.
- These protocols reside in the Data Link Layer and in the Transport Layer of the OSI (Open Systems Interconnection) reference model.
- They are named so because they provide for automatic retransmission of frames that are corrupted or lost during transmission.
- ARQ is also called Positive Acknowledgement with Retransmission (PAR).

ARQs are used to provide reliable transmissions over unreliable upper layer services. They are often used in Global System for Mobile (GSM) communication.

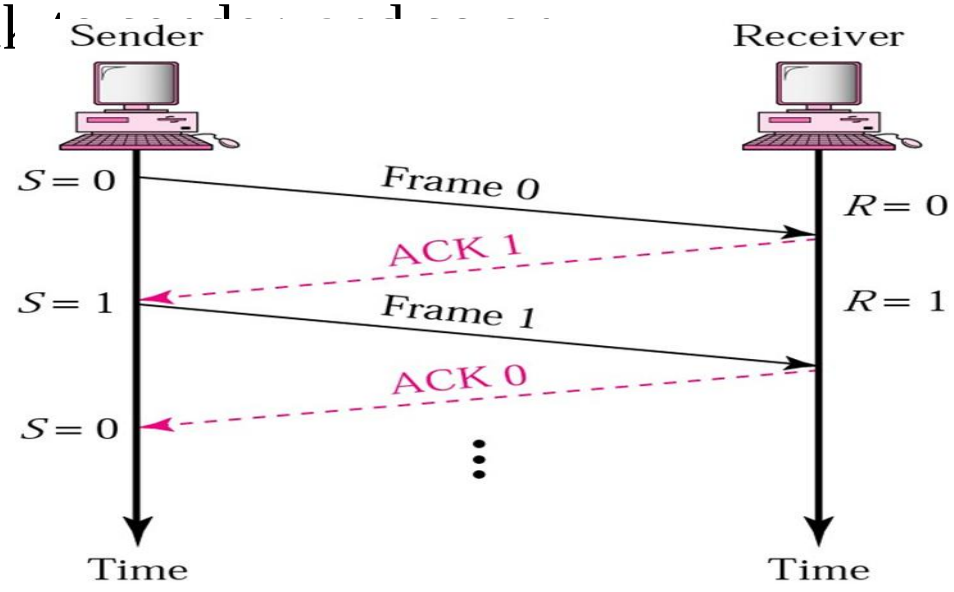
Working Principle

- In these protocols, the receiver sends an acknowledgement message back to the sender if it receives a frame correctly.
- If the sender does not receive the acknowledgement of a transmitted frame before a specified period of time, i.e. a timeout occurs, the sender understands that the frame has been corrupted or lost during transit.
- So, the sender retransmits the frame. This process is repeated until the correct frame is transmitted.

Noisy Channels

Stop-and-wait ARQ

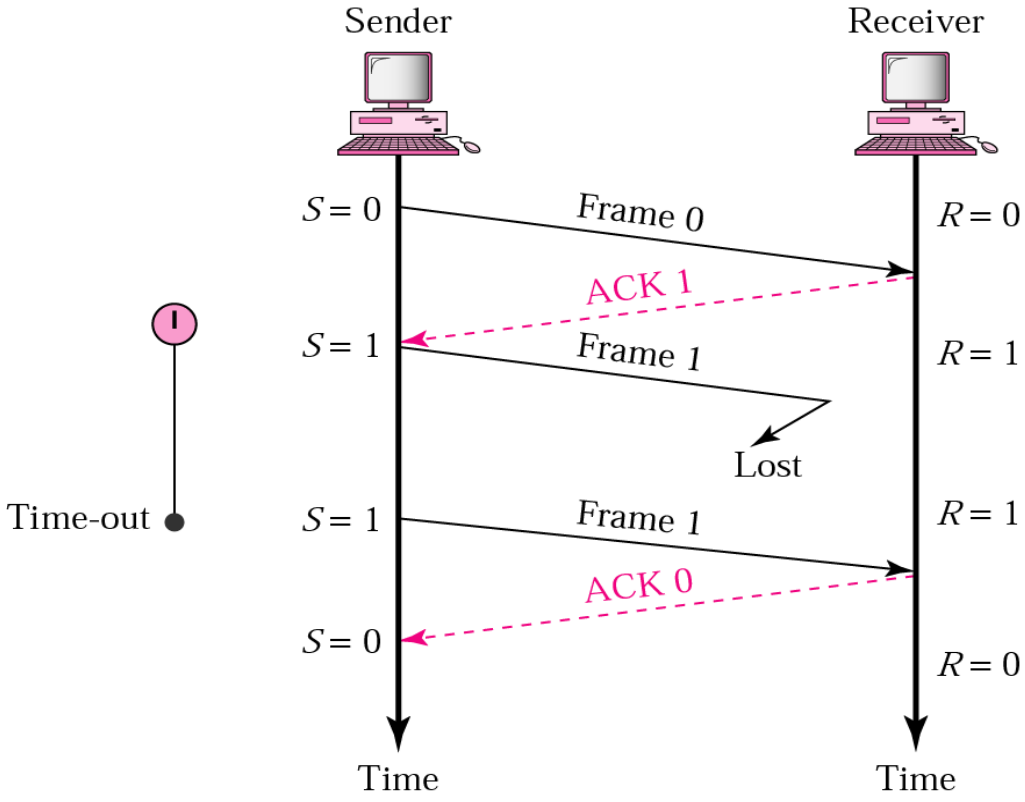
- Stop-and-wait ARQ is the most basic mechanism among three, and it is the foundation of the other two.
- In the stop and wait ARQ, frame and ACK are numbered 0 and 1 alternately.
- Frames 0 sends to receiver, ACK 1 will be sent back to sender; frame 1 goes to receiver, ACK 0 will be bacl



Stop-and-wait ARQ normal operation

Stop-and-wait ARQ, lost frame

As the figure shows, from sender to receiver, frame 1 is lost, but sender is still expecting the ACK 0 back. After timer time out, frame 1 sends again.

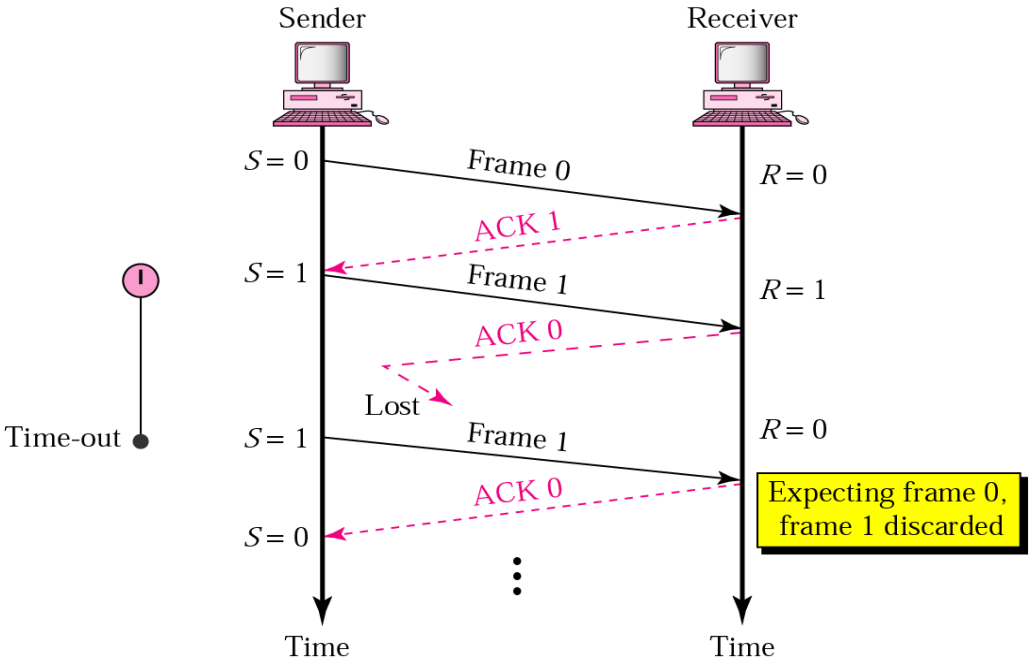


Stop-and-wait ARQ lost frame

Lost or delayed ACK

Stop-and-wait ARQ, lost ACK frame

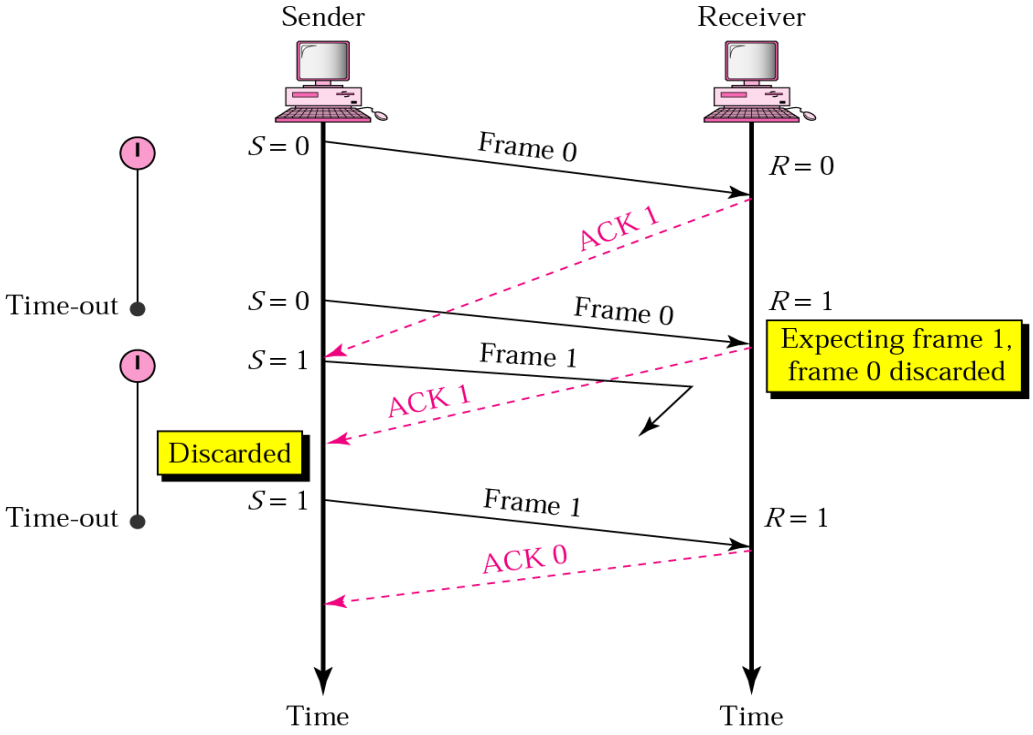
As the figure shows, when ACK 0 sends back to sender, this frame has lost. So after time out, frame 1 sends again. Receiver side is excepting for frame 0, so frame 1 is discarded.



Stop-and-wait ARQ lost ACK

Stop-and-wait ARQ, delayed ACK

Frame 0 sends to receiver, ACK 1 sends back to Sender, but the transmission has delayed. After time out, sender sends frame 0 again; receiver is excepting frame 1. Therefore, the frame 0 is discarded, ACK 1 is discarded. Frame 1 sends again.



Stop-and-wait ARQ delayed ACK

Go-Back-N ARQ

- If the link quality is very good (assume there is no error frame), for Stop-and-wait ARQ, each time only one frame can be transmitted.
- But in reality, the link is always noisy, frame lost or damaged happens. The sender and receiver will have nothing to do except waiting.
- Therefore the Stop-and-wait ARQ is very low efficiency. To improve to this, we can send several frames insides only one. Therefore the Go-Back-N ARQ has been generated.

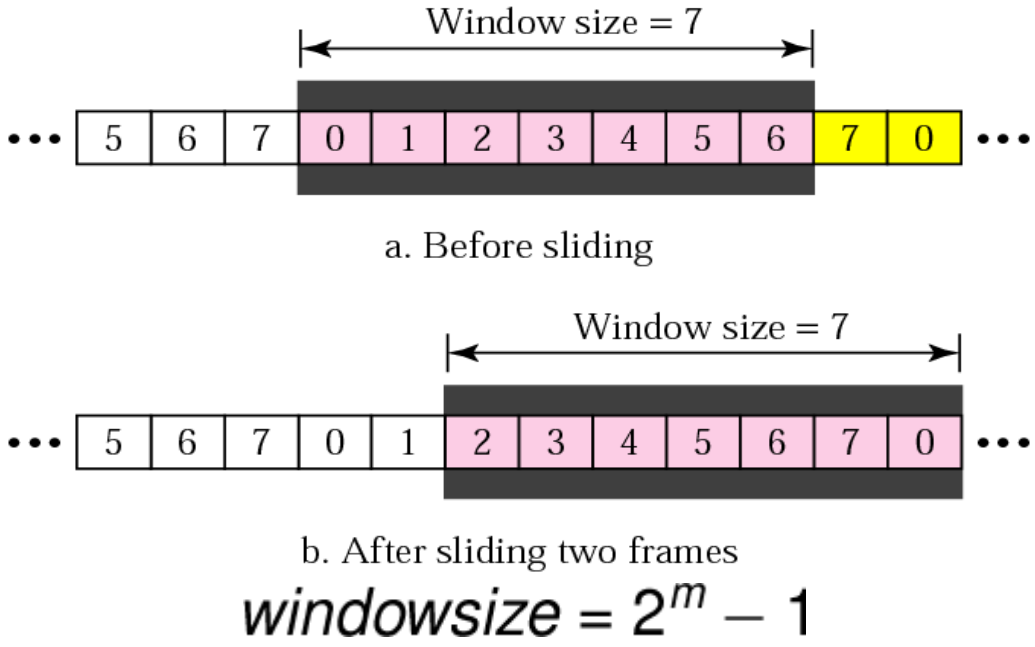
Sequence Numbers

- Because each time, there are several frames waiting for transmission, so we must number them sequentially.
- We set a limit for the frames. If the header of frame allows m bits for the sequence number, those frames range will be 0 to 2^m-1.
- If m=2, sequence number will be 0,1,2,3,0,1,2,3… repeat in this way. Unlike Stop- and-wait ARQ's 0,1,0,1…

Sender sliding window

- Now, a "group" of frames send to receiver, we need something to hold this "group" until ACK arrived.
- Next, the concept of "sliding window" is introduced. The window size is fixed which is 2^m-1.

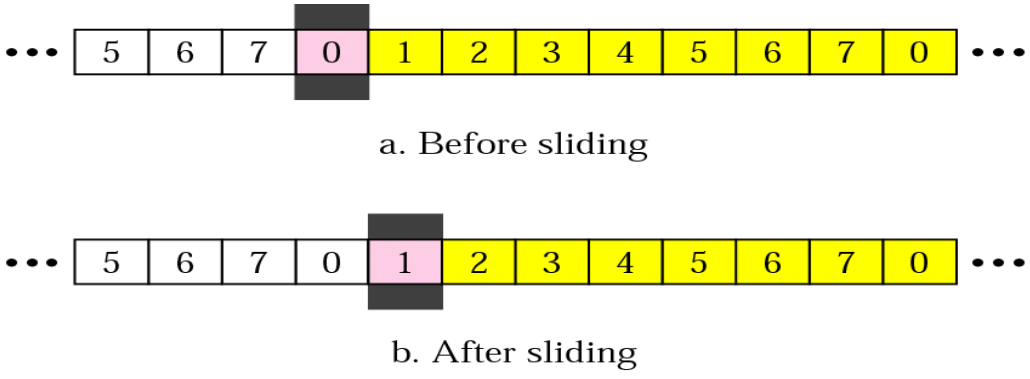
- Inside this sliding window, there are the copies of the transmission frames. When the correct ACK arrived, sliding window will slid forward.



Go-Back-N ARQ sender sliding window

Receiver sliding window

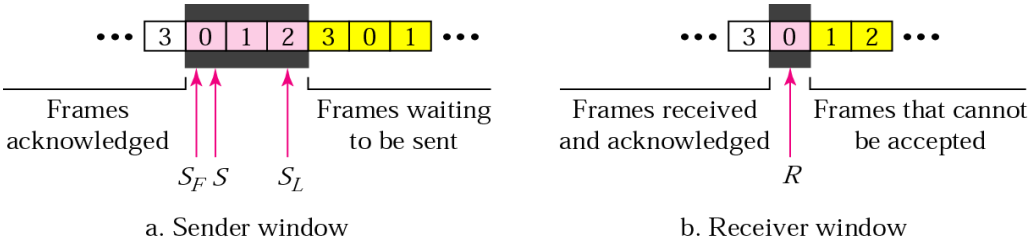
Receiver sliding window in Go-Back-N ARQ is always 1. It's waiting for the correct frame comes in correct order, then sends back the ACK and slide forward. If the frame is lost or damaged, receiver will wait for the resend. Even the rest of the frame is correct, receiver will discard them automatically.



Go-Back-N ARQ receiver sliding window

Control Variables

In the Go-Back-N ARQ, sender's control variables are S, S_F, S_L. But receiver's variable is still R. Slide window size is W. S is the sequence number of latest sent frame, S_F is the sequence number of the first frame in the slide window, S_L is the sequence number of the last frame in the slide window. R is the sequence number of the excepted frame. $W=S_L-S_F+1=2^m-1$. Only when R and sequence number of received frame are matched, frame accept, otherwise discard it.



Go-Back-N ARQ control variable

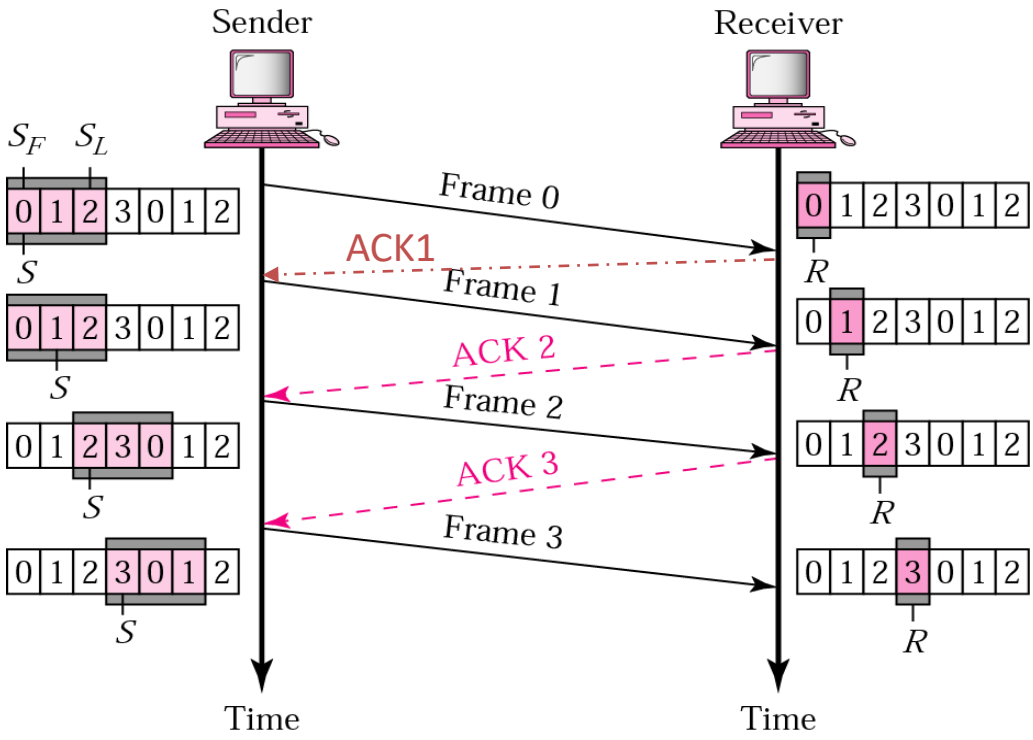
Timer

Inside of slide window, each sent frame has individual timer. The total timer number is equal to the slide window size.

Lost or damaged frame

- Receiver will send back an ACK to sender if the correct frame received (right frame in right order).
- If the frame is lost or damaged, receiver will remain silence.
- If there is no ACK back not back on time, sender will resend group of frames, from S to S_L.
- The receiver is only "loyalty" to the first incoming frame, no matter what are the conditions of the following frame, even they are correct, "ignore" them still.

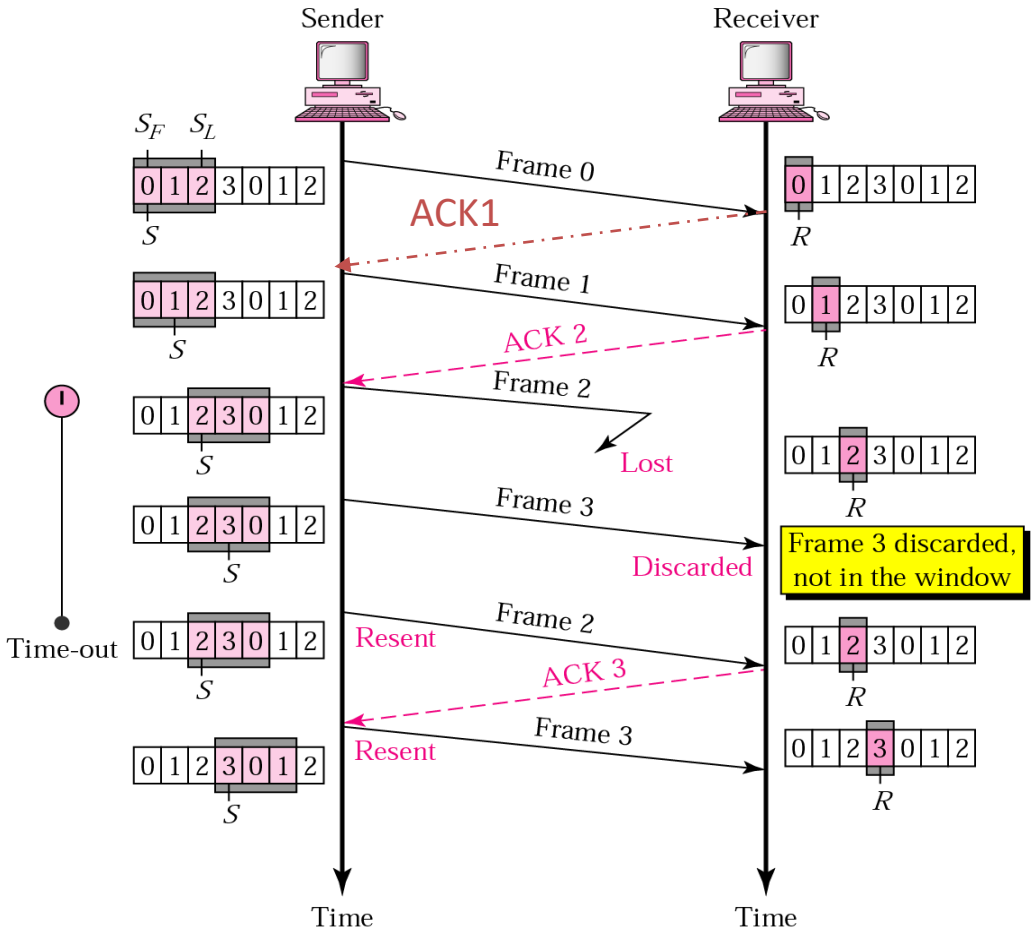
Go-Back-N ARQ, normal operation



Go-Back-N ARQ normal operation

Frame 0 & 1 send, ACK 1 & 2 back to sender. Frame 2 send, ACK 3 send back. Then frame 3 send to receiver.

Go-Back N ARQ, lost frame



Go-Back-N ARQ lost frame

Frame 0 & 1 send, ACK 1 & 2 back to sender. Frame 2 & 3 send, but frame 2 lost in the transmission. When frame 3 received out of order, this frame 3 will be discarded by receiver. After time out, frame 2 resent, then receiver send ACK 3 back and then frame 3 resent.

Selective Repeat ARQ

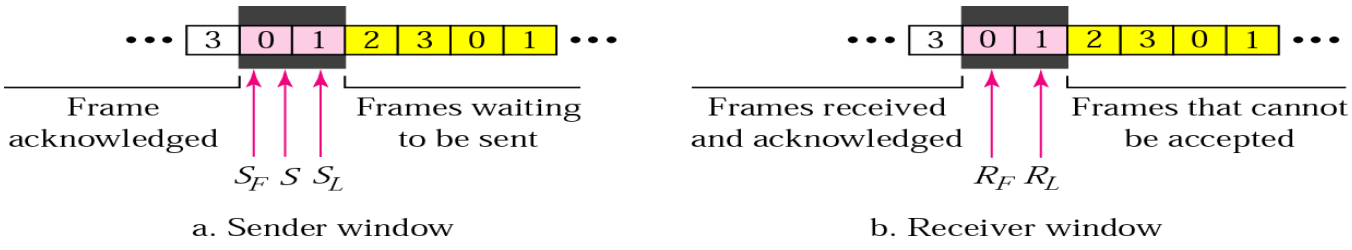
- In reality, links usually noisy. Lost or damaged frame occurred very often. This means lots of frames will be discard by receiver automatically, even they are correct only with wrong order.
- The sender has to send those "innocent" frames again and again. Bandwidth has been used up, transmission speed slow down.
- All of this, made Go-Back-N ARQ is still not very high efficiency. Therefore the advanced version protocol came out. Not need to resend N frames. Instead, only the lost or damaged frame resend, this mechanism is called Selective Repeat ARQ.

Sender slide window

The control variables in Selective Repeat ARQ are same as in Go-Back-N ARQ: S_F, S_L and S. But the slide window size changed into 2^{m-1}.

Receiver slide window

Receiver has 2 control variables, R_F and R_L. The slide window size also changed into 2^{m-1}.



Selective Repeat ARQ receiver slide window

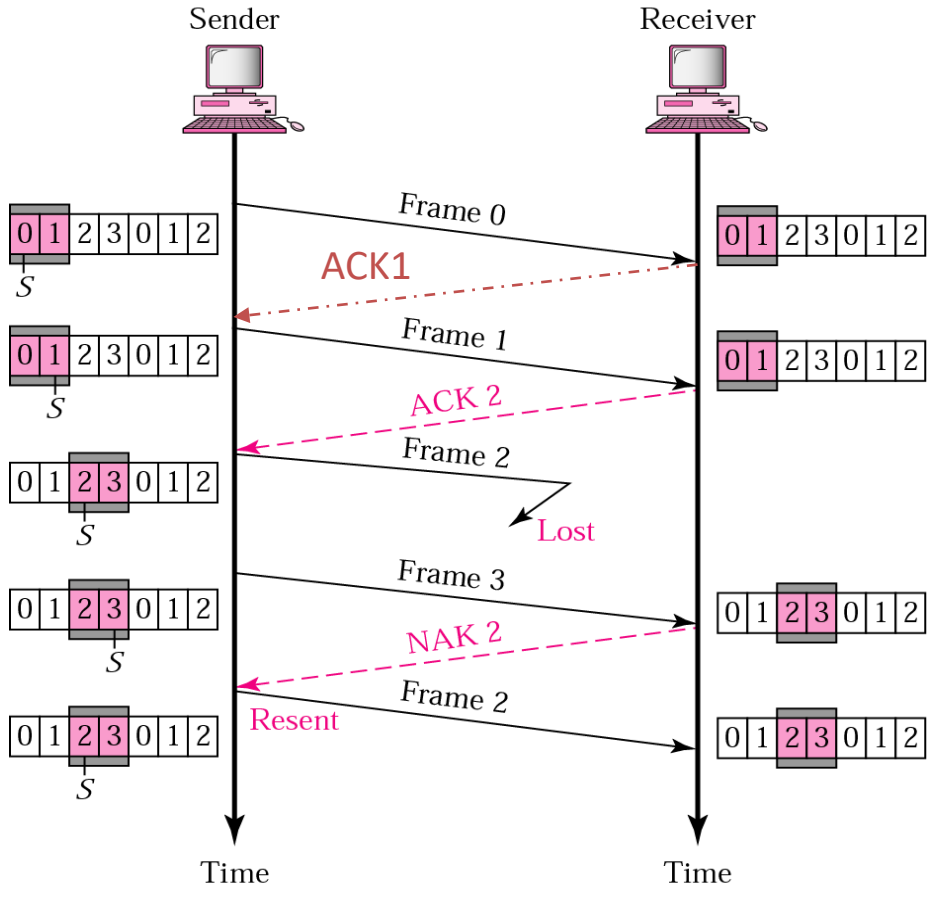
Now, the sender slide window size is same as receiver's. In the Go-Back-N ARQ, each time receiver search one specific sequence number; in Selective Repeat

ARQ, each time receiver look for a range specific sequence number, from R_F to R_L.

NAK

NAK means negative acknowledgment, it does only exist in Selective Repeat ARQ.

Selective Repeat ARQ, lost frame



Selective Repeat ARQ lost frame

SUMMARY AND CONCLUSION

Comparing three different ARQ protocols-

	Stop-and-Wait	Go-back-N	Selective Repeat
Sender slide window size	1	2^m-1	2^{m-1}
Receiver slide window size	1	1	2^{m-1}
ACK	Yes	Yes	Yes
NAK	No	No	Yes
Frame sequence	0,1,0,1...	$0...2^m-1$	$0...2^m-1$
Bandwidth utilization	Low	Medium	high