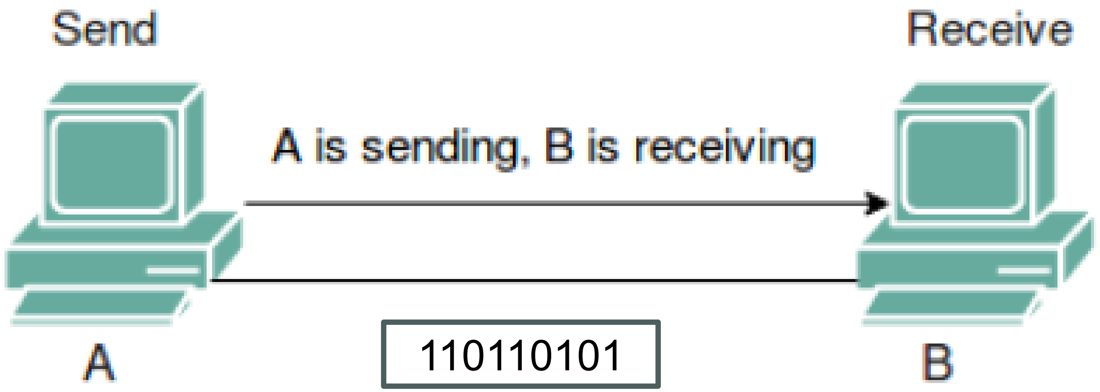
Data Communication and Computer Networks

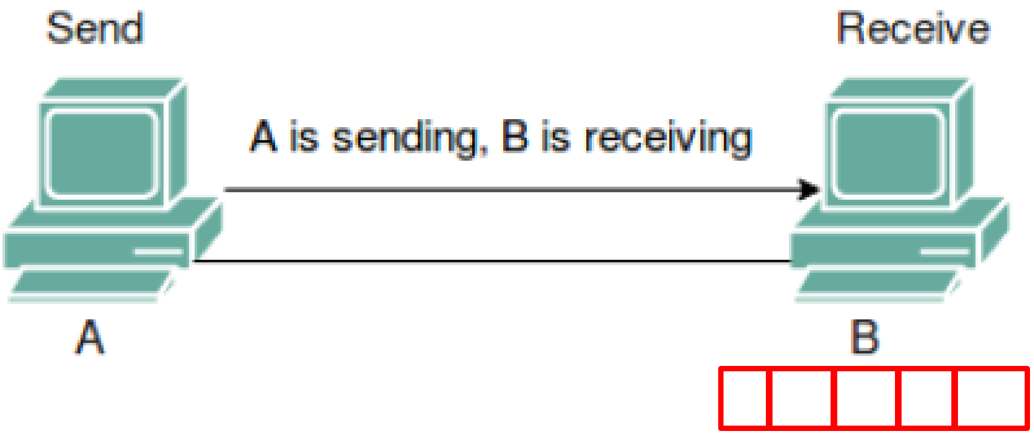
FLOW CONTROL

- Flow control is a technique for assuring that a transmitting entity does not overwhelm a receiving entity with data.



- The data are sent in a sequence of frames.
- Each frame contains a portion of the data and some control information.

FLOW CONTROL



- Receiving entity typically allocates a data buffer of some maximum length for a transfer.

- In the absence of flow control, the receiver's buffer may fill up and overflow while it is processing old data.

FLOW CONTROL

- Mechanisms for flow control in the absence of errors.
- Figure shows vertical-time sequence diagram.

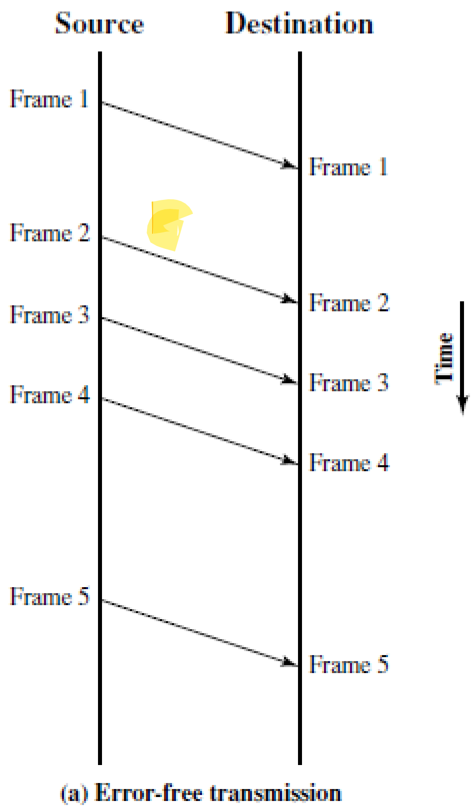


Figure 7.1 Model of Frame Transmission

FLOW CONTROL

- Transmission time : Time it taken for a station to emit all of the bits of a frame onto the medium.

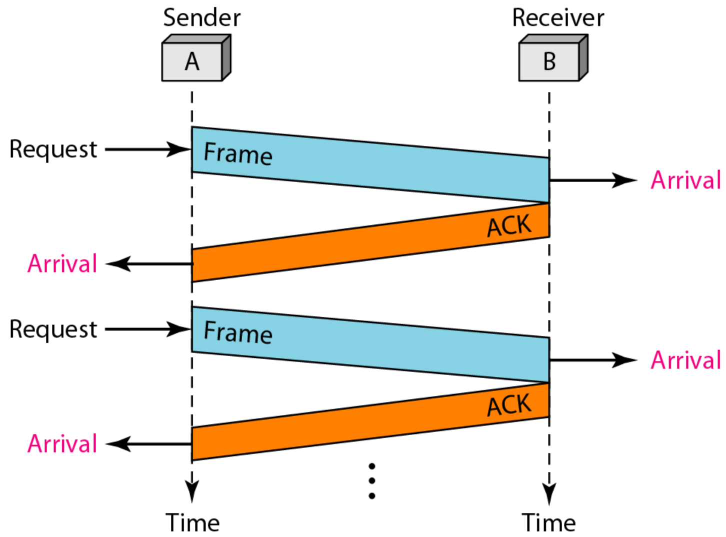
- Propagation time : Time taken for a bit to traverse the link between source and destination.

FLOW CONTROL

assumption

- All frames that are transmitted are successfully received.
- Frames arrive in the same order in which they are sent.
- However, each transmitted frame suffers an arbitrary and variable amount of delay before reception.

Stop-and-Wait Flow Control



- A source entity transmits a frame.
- The source must wait until it receives the acknowledgment before sending the next frame.
- The destination can thus stop the flow of data simply by with holding acknowledgment.

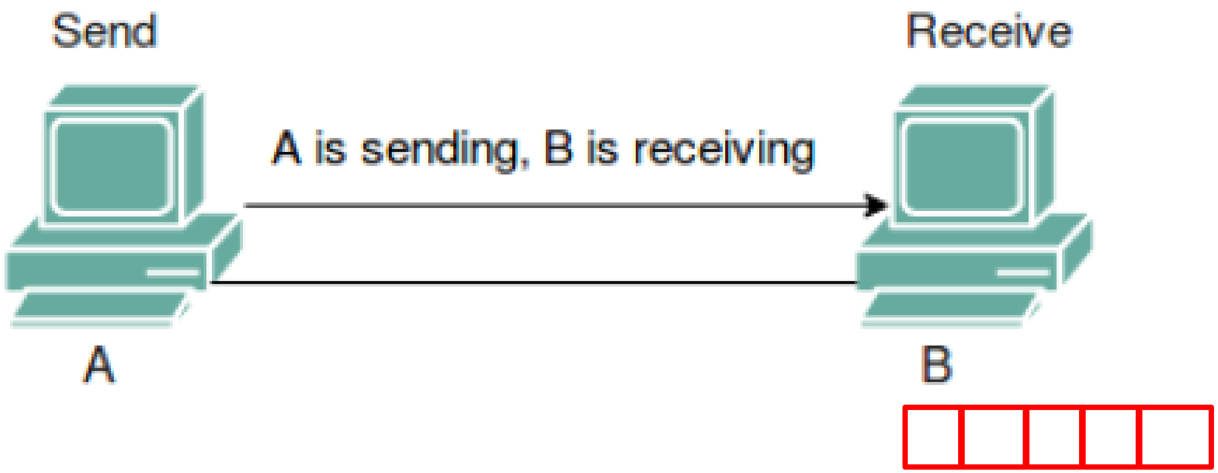
Sliding-Window Flow Control

- The problem with stop and wait is that only one frame at a time can be in transit.

- For very high data rates, for very long distances between sender and receiver, stop-and-wait flow control provides inefficient line utilization.

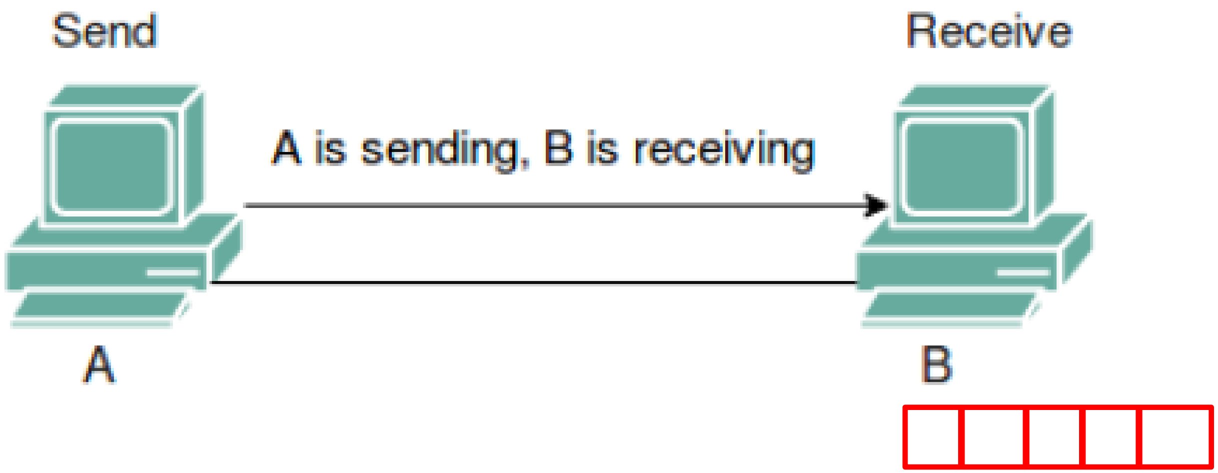
- Efficiency can be greatly improved by allowing multiple frames to be in transit at the same time.

Sliding-Window Flow Control



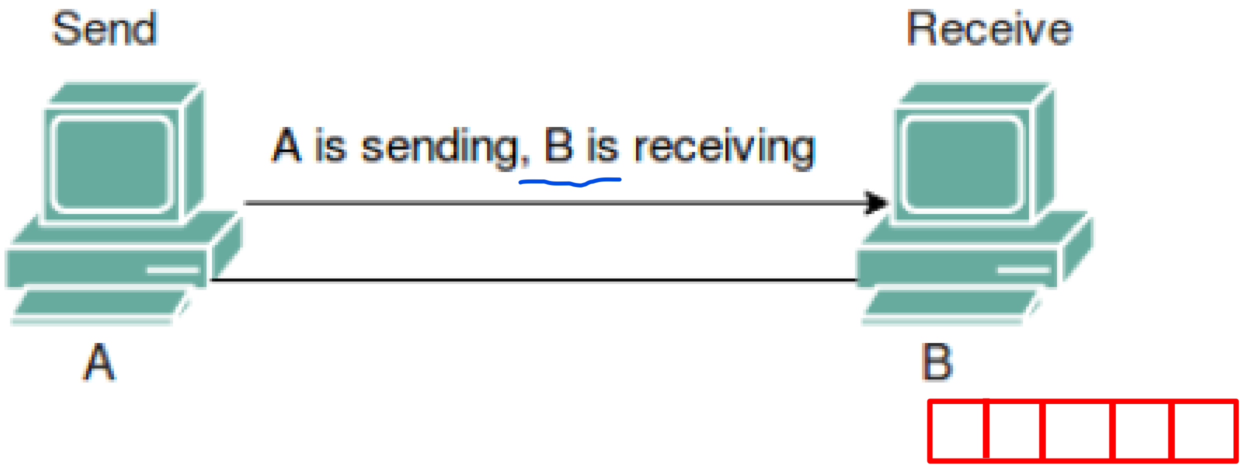
- Consider two stations A and B.
- Station B allocates buffer space for *W* frames.
- Thus, B can accept *W* frames, and A is allowed to send *W* frames without waiting for any acknowledgments.

Sliding-Window Flow Control



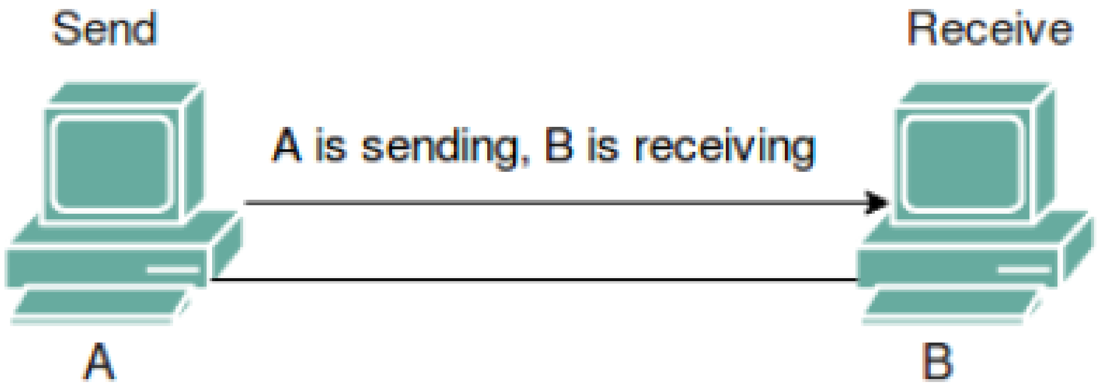
- To keep track of which frames have been acknowledged, each is labeled with a sequence number.
- B acknowledges a frame by sending an acknowledgment that includes the sequence number of the next frame expected.

Sliding-Window Flow Control



- This scheme can also be used to acknowledge multiple frames.
- B could receive frames 2, 3, and 4 but withhold acknowledgment until frame 4 has arrived.
- By then returning an acknowledgment with sequence number 5, B acknowledges frames 2, 3, and 4 at one time.

Sliding-Window Flow Control



- Sender maintains a list of sequence numbers that it is allowed to send. (sender window)
- Receiver maintains a list of sequence numbers that it is prepared to receive. (Receiver window)
- Each of these lists can be thought of as a *window* of frames.
- The operation is referred to as **sliding-window flow control**.

Sliding-Window Flow Control

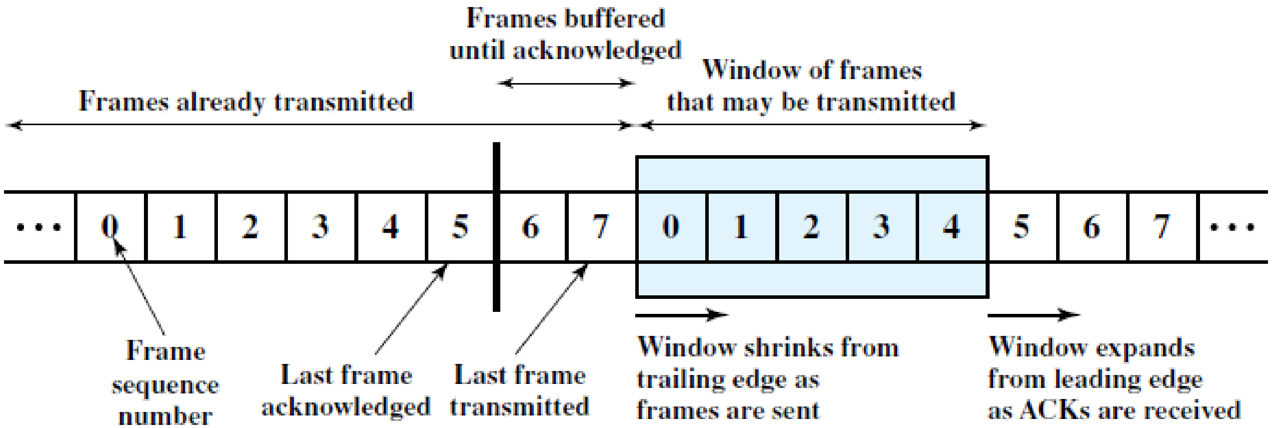
- Because the sequence number to be used occupies a field in the frame, it is limited to a range of values.

- For a 3-bit field, the sequence number can range from 0 to 7.

- For a *k*-bit field the range of sequence numbers is 0 through $2^k - 1$,

- The maximum window size is $2^k - 1$,

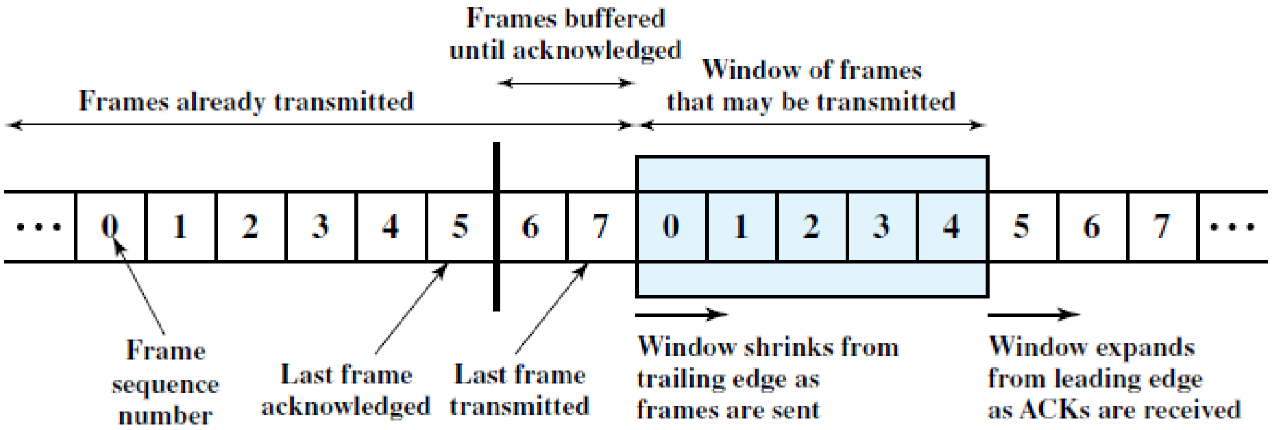
Sliding-Window Flow Control



(a) Sender's perspective

- Assumes the use of a 3-bit sequence number
- The shaded rectangle indicates the frames that may be sent;
- Here window size is 5.

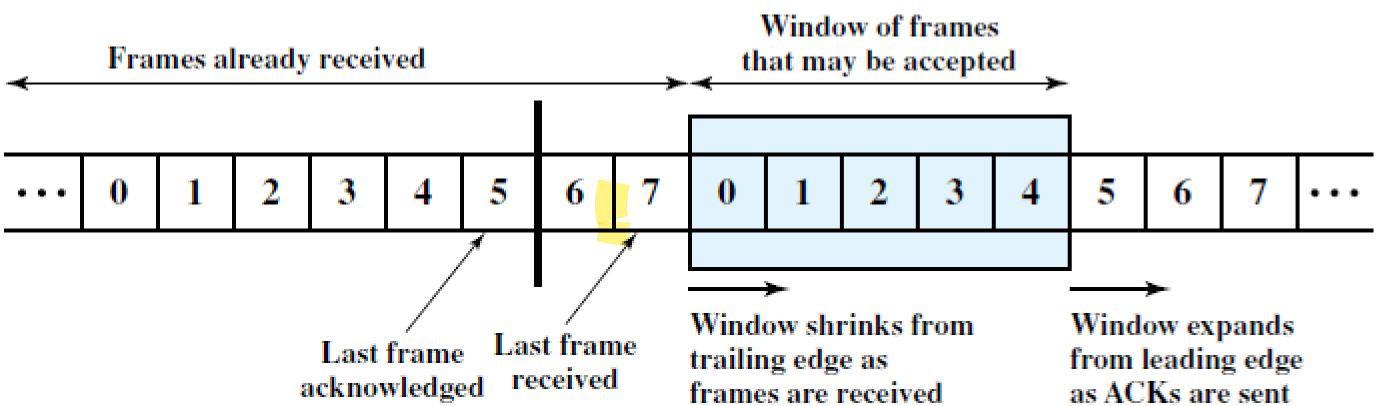
Sliding-Window Flow Control



(a) Sender's perspective

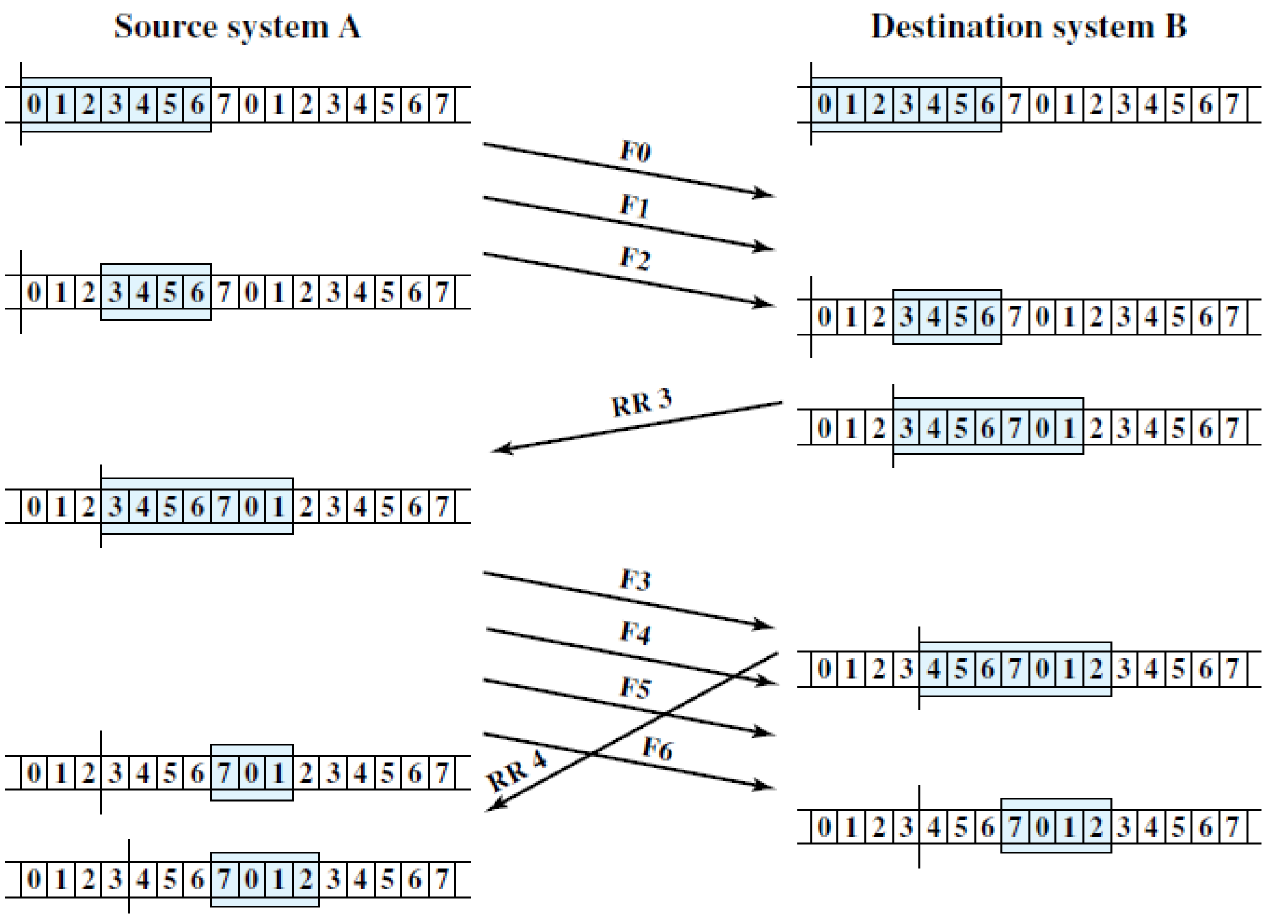
- Each time a frame is sent, the shaded window shrinks;
- each time an acknowledgment is received, the shaded window grows.
- Frames between vertical bar and the shaded window have been sent but not yet acknowledged. Sender must buffer these frames in case they need to be retransmitted.

Sliding-Window Flow Control



(b) Receiver's perspective

Sliding-Window Flow Control



- Example assumes a 3-bit sequence number field and a maximum window size of seven frames.

Sliding-Window Flow Control

- Receive Not Ready (RNR) message acknowledges former frames but forbids transfer of future frames.
- RNR 5 means “received all frames up to number 4 but unable to accept any more at this time.”
- At some subsequent point, the station must send a normal acknowledgment to reopen the window.

Sliding-Window Flow Control

- So far discussed : transmission in one direction only.
- If two stations exchange data, each needs to maintain two windows, one for transmit and one for receive.
- To provide efficient support for this requirement, a feature known as **piggybacking** is typically provided.
- Each **data frame** includes a field that holds the sequence number of that frame plus a field that holds the sequence number used for acknowledgment.

Sliding-Window Flow Control

- if a station has data and acknowledgment to send, it sends both together in one frame.
- if a station has an acknowledgment but no data to send, it sends a separate **acknowledgment frame**, such as RR or RNR.
- If a station has data to send but no acknowledgment to send, it must repeat the last acknowledgment sequence number that it sent.
  - This is because the data frame includes a field for the acknowledgment number, and some value must be put into that field.
  - When a station receives a duplicate acknowledgment, it simply ignores it.

ERROR CONTROL

- Error control refers to mechanisms to detect and correct errors that occur in the transmission of frames.

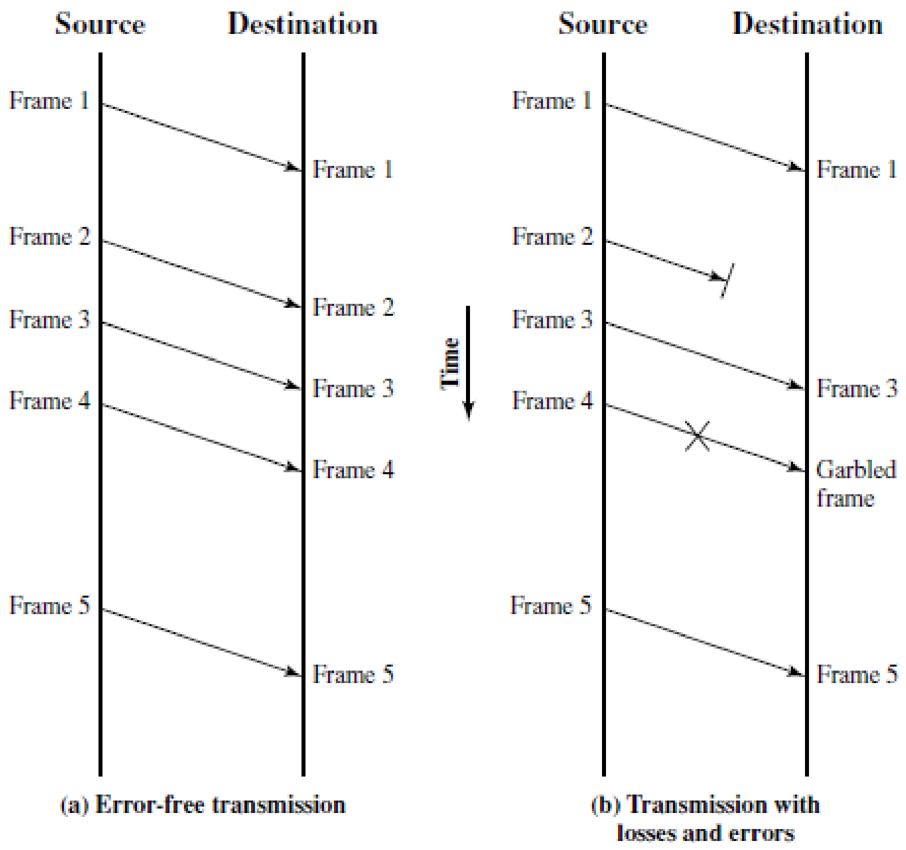


Figure 7.1 Model of Frame Transmission

ERROR CONTROL

The possibility of two types of errors:

- **Lost frame:** A frame fails to arrive at the other side.
- **Damaged frame:** Some of the bits are in error.

ERROR CONTROL

Most common techniques for error control are :

- **Error detection:** discussed.

- **Positive acknowledgment:** Destination returns a positive acknowledgment to successfully received, error-free frames.

- **Retransmission after timeout:** Source retransmits a frame that has not been acknowledged after a predetermined amount of time.

- **Negative acknowledgment and retransmission:** Destination returns a negative acknowledgment to frames in which an error is detected. Source retransmits such frames.

ERROR CONTROL

Collectively, these mechanisms are all referred to as **automatic repeat request** (ARQ);

The effect of ARQ is to turn an **unreliable data link into a reliable one.**

Three versions of ARQ have been standardized:

- Stop-and-wait ARQ
- Go-back-N ARQ
- Selective-reject ARQ

All of these forms are based on the use of the flow control techniques.

Stop-and-Wait ARQ

- Stop-and-wait ARQ is based on the stop-and-wait flow control technique.

  - Source station transmits a single frame and then must await an acknowledgment (ACK).

  - No other data frames can be sent until the destination station's reply arrives.

Stop-and-Wait ARQ

- Two sorts of errors could occur.
- First, the frame that arrives at the destination could be damaged.
- Receiver detects this by error-detection technique & simply discards the frame.

- The source station is equipped with a timer.
- After a frame is transmitted, the source station waits for an acknowledgment.
- If no acknowledgment is received by the time that the timer expires, then the same frame is sent again.
- Requires that the transmitter maintain a copy of a transmitted frame until an acknowledgment is received for that frame.

Stop-and-Wait ARQ

- The second sort of error is a damaged acknowledgment.
- Station A sends a frame. Frame is received correctly by station B, which responds with an acknowledgment (ACK).
- ACK is damaged in transit and is not recognizable by A, which will therefore time out and resend the same frame.

Stop-and-Wait ARQ

- This duplicate frame arrives and is accepted by B.

- B has therefore accepted two copies of the same frame as if they were separate.

- To avoid this problem, frames are alternately labeled with 0 or 1, and positive acknowledgments are of the form ACK0 and ACK1.

- ACK0 acknowledges receipt of a frame numbered 1 and indicates that the receiver is ready for a frame numbered 0.

Stop-and-Wait ARQ

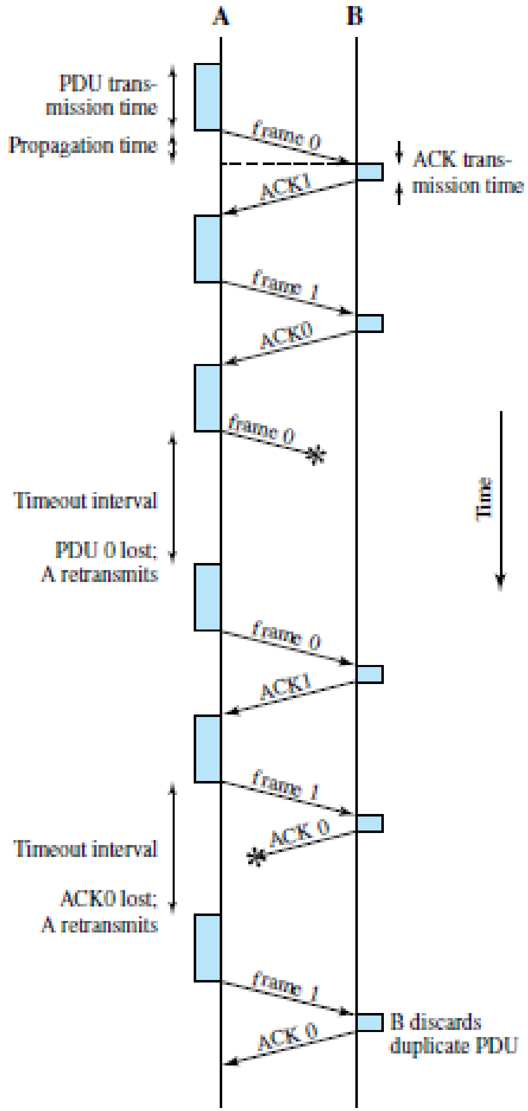


Figure 7.5 Stop-and-Wait ARQ

- Frame transmitted by A is lost or damaged and therefore B does not return an ACK.
- A times out and retransmits the frame.

- A transmits a frame labeled 1 but the ACK0 for that frame is lost.
- A times out and retransmits the same frame.
- When B receives two frames in a row with the **same label**, it discards the second frame but sends back an ACK0 to each.

Go-Back-N ARQ

- Based on sliding-window flow control that is most commonly used.

- Station send a series of frames sequentially numbered modulo some maximum value.

Go-Back-N ARQ

no errors :

- Destination will acknowledge incoming frames (RR = receive ready, or piggybacked acknowledgment).

error

- send a negative acknowledgment (REJ = reject) for that frame.

- Destination station will discard that frame and all future incoming frames until the frame in error is correctly received.

- Source station, when it receives a REJ, must retransmit the frame in error **plus** all succeeding frames that were transmitted in the interim.

Go-Back-N ARQ

- Suppose that station A is sending frames to station B.
- After each transmission, A sets an acknowledgment timer for the frame just transmitted.
- Suppose that B has previously successfully received frame (i – 1) and A has just transmitted frame *i.*
- The go-back-N technique takes into account the following contingencies:

$(i - 1)$

Go-Back-N ARQ

1. **Damaged frame.** If the received frame is invalid (i.e., B detects an error), B discards the frame. There are two subcases:

   a) Within a reasonable period of time, A subsequently sends frame $(i + 1)$. ,B receives frame $(i + 1)$. out of order and sends a REJ *i.* A must retransmit frame *i* and all subsequent frames.

   a) A does not soon send additional frames. B receives nothing and returns neither an RR nor a REJ. When A's timer expires, it transmits an RR frame that includes a bit known as the P bit, which is set to 1. B interprets the RR frame with a P bit of 1 as a command that must be acknowledged by sending an RR indicating the next frame that it expects, which is frame *i.* When A receives the RR, it retransmits frame *i.* Alternatively, A could just retransmit frame *i* when its timer expires.

Go-Back-N ARQ

2. Damaged RR. There are two subcases:

a) B receives frame *i* and sends RR (i+1) which suffers an error in transit. Because acknowledgments are cumulative (e.g., RR 6 means that all frames through 5 are acknowledged), it may be that A will receive a subsequent RR to a subsequent frame and that it will arrive before the timer associated with frame *i* expires.

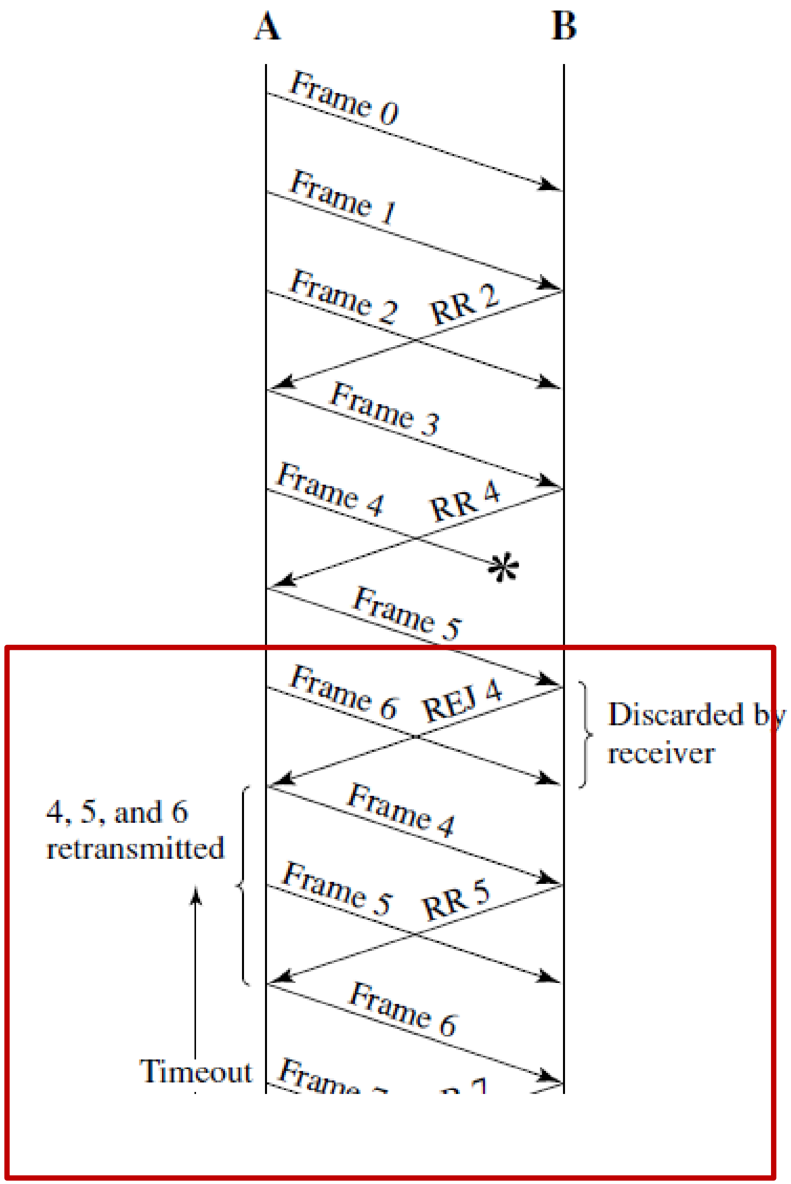
b) If A's timer expires, it transmits an RR command as in Case 1b. It sets another timer, called the P-bit timer. If B fails to respond to the RR command, or if its response suffers an error in transit, then A's P-bit timer will expire. At this point, A will try again by issuing a new RR command and restarting the P-bit timer. This procedure is tried for a number of iterations. If A fails to obtain an acknowledgment after some maximum number of attempts, it initiates a reset procedure.

3. Damaged REJ. If a REJ is lost, this is equivalent to Case 1b.

$(i + 1)$

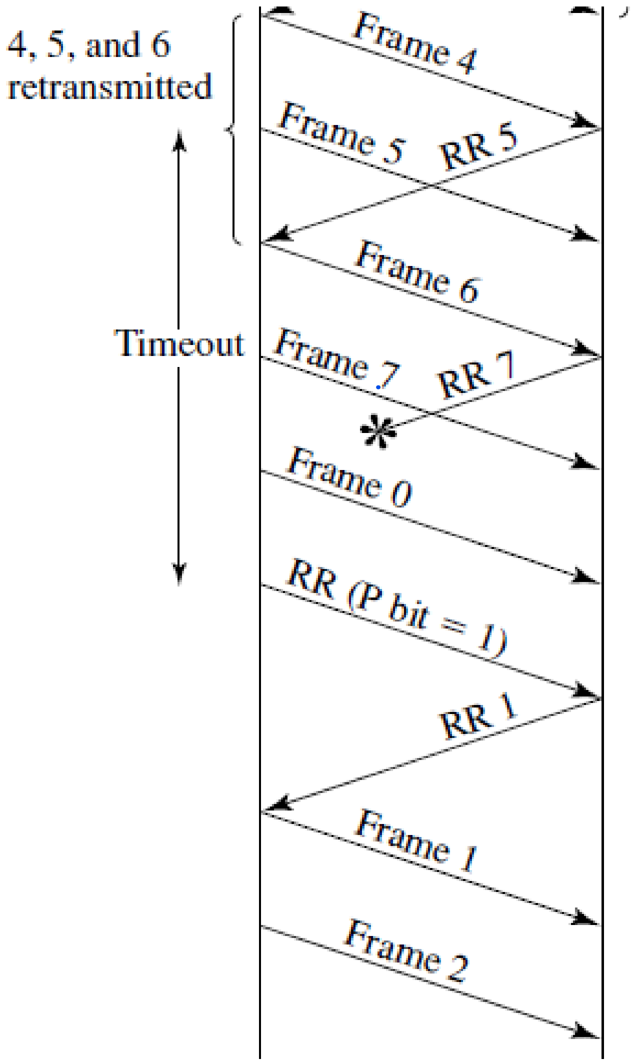
Go-Back-N ARQ

- Here, frame 4 is damaged.
- Frames 5 and 6 are received out of order and are discarded by B.
- When frame 5 arrives, B immediately sends a REJ 4.
- When the REJ to frame 4 is received, not only frame 4 but frames 5 and 6 must be retransmitted.



Go-Back-N ARQ

- Example of retransmission after timeout.
- No acknowledgment is received for frame 5 within the timeout period, so A issues an RR to determine the status of B.



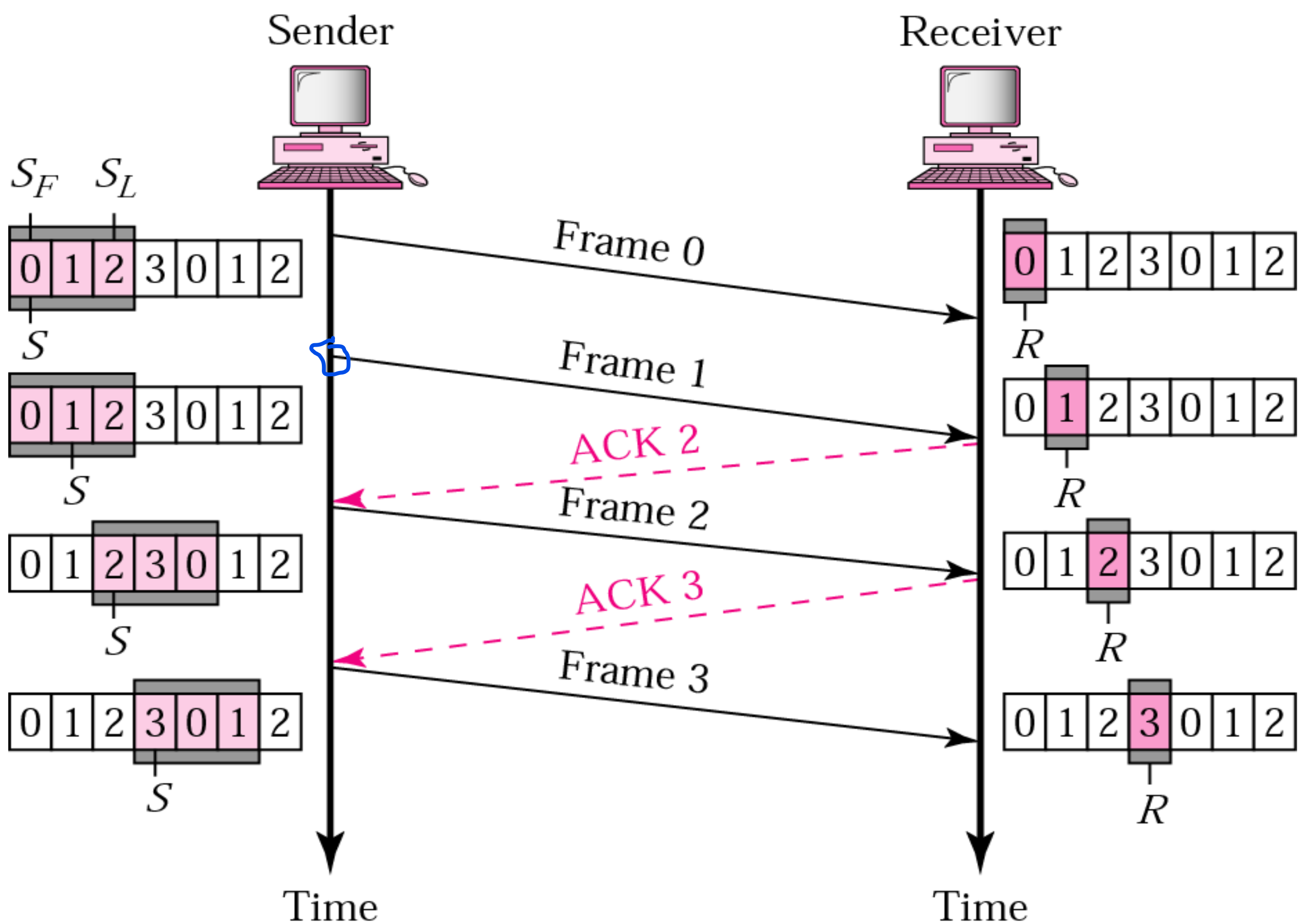
Go-Back-N ARQ

- For a *k*-bit sequence number field, which provides a sequence number range of 2^k, the maximum sender window size is limited to $2^k - 1$
- Receiver window size is 1.

Go-Back-N ARQ

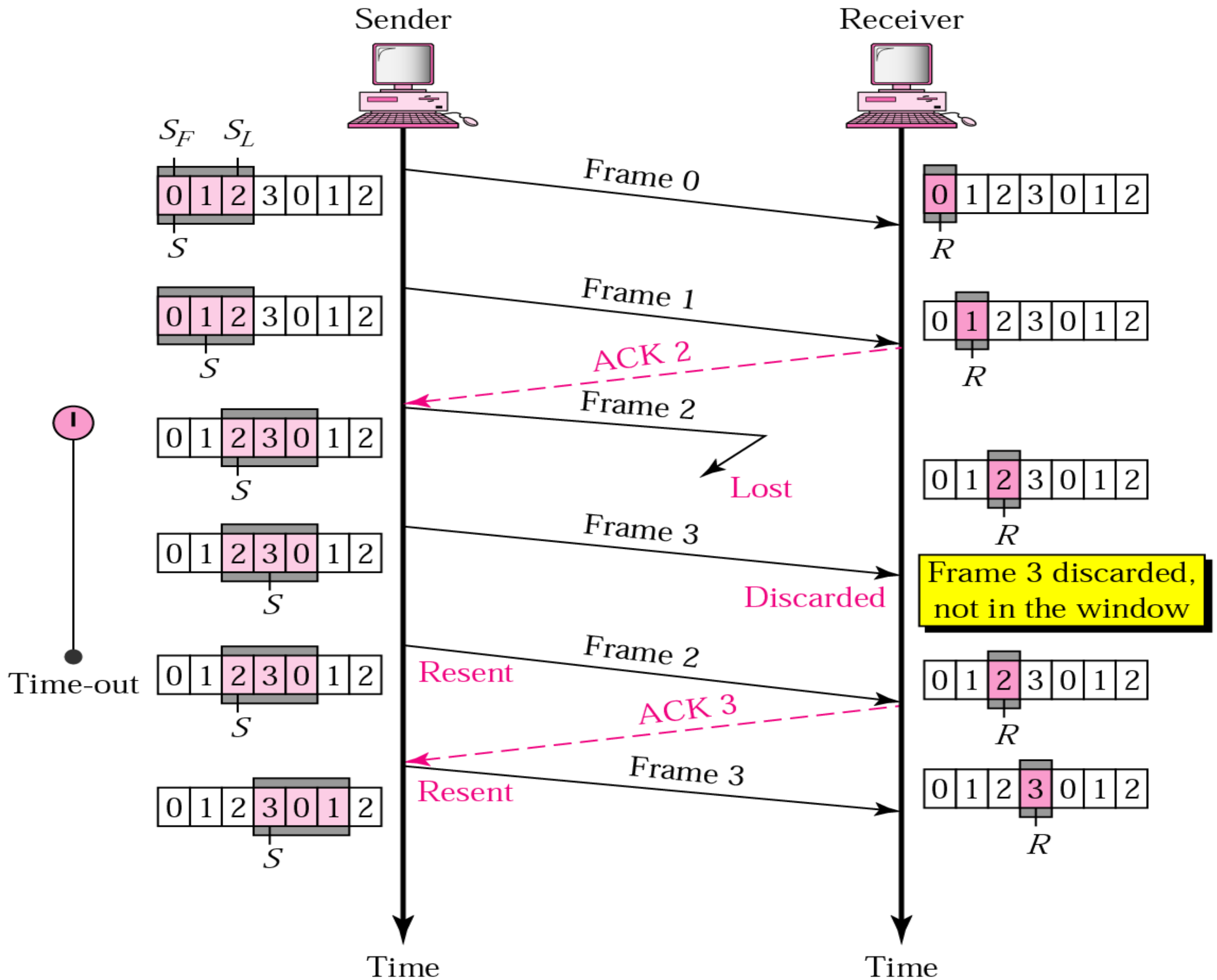
- Assume a 3-bit sequence number (sequence number space = 8).
- Suppose a station sends frame 0 and gets back an RR 1 and then sends frames 1, 2, 3, 4, 5, 6, 7, 0 and gets another RR 1.
- This could mean that all eight frames were received correctly and the RR 1 is a cumulative acknowledgment.
- It could also mean that all eight frames were damaged or lost in transit, and the receiving station is repeating its previous RR 1.
- The problem is avoided if the maximum window size is limited to $\boxed{7\ (2^3 - 1)}$.

Go-Back-N ARQ, normal operation

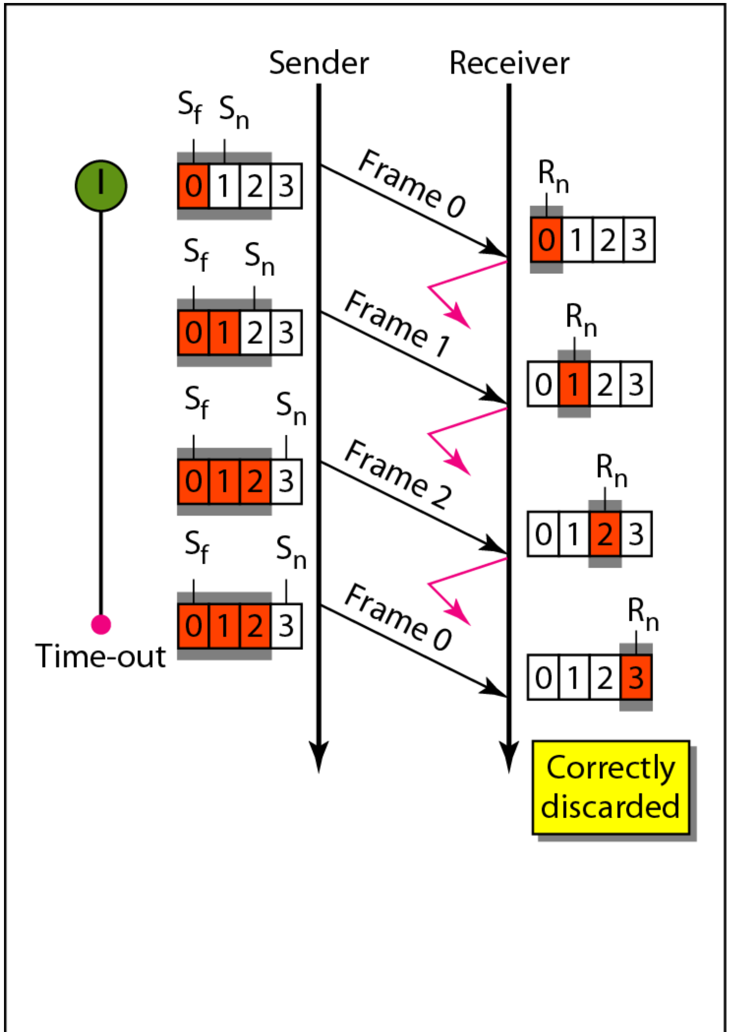


Sender window will slide to frame number (i)
when (ACK i) is received.

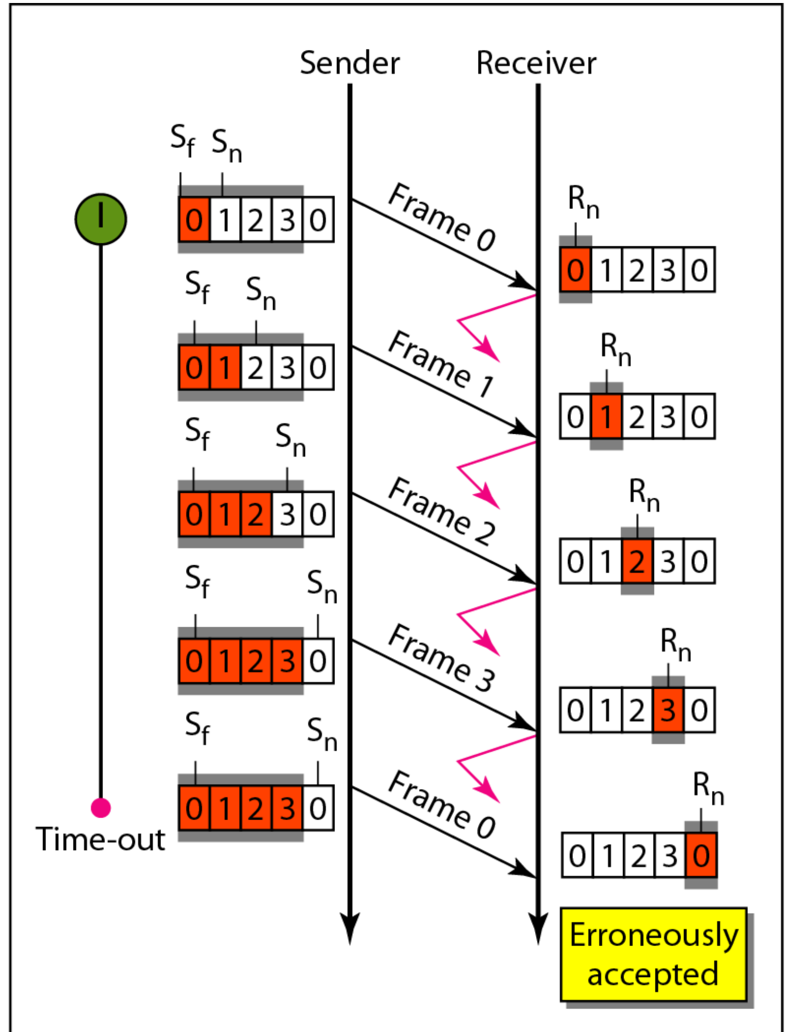
Go-Back-N ARQ, lost frame



Window size for Go-Back-N ARQ



a. Window size < 2^m



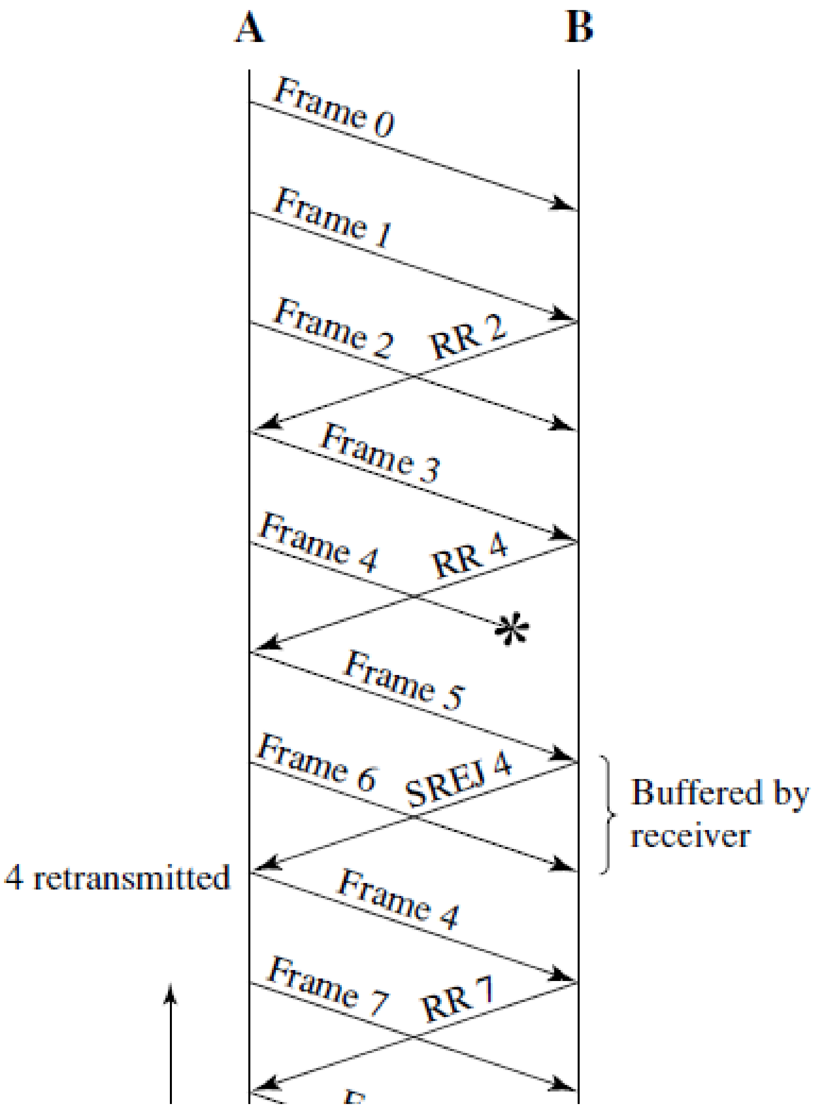
b. Window size = 2^m

Selective-Reject ARQ

With selective-reject ARQ, the only frames retransmitted are those that receive a negative acknowledgment, in this case called SREJ, or those that time out.

Selective-Reject ARQ

- When frame 5 is received out of order, B sends a SREJ 4, indicating that frame 4 has not been received.
- However, B continues to accept incoming frames and buffers them until a valid frame 4 is received.
- At that point, B can place the frames in the proper order for delivery to higher-layer software.



Selective-Reject ARQ

- Selective reject would appear to be more efficient than go-back-N, because it minimizes the amount of retransmission.

- On the other hand, the receiver :
  - must maintain a buffer large enough to save post-SREJ frames until the frame in error is retransmitted and
  - must contain logic for reinserting that frame in the proper sequence.

- The transmitter, too,
  - requires more complex logic to be able to send a frame out of sequence.

Selective-Reject ARQ

- Because of such complications, select-reject ARQ is much less widely used than go-back-N ARQ.
- Selective reject is a useful choice for a satellite link because of the long propagation delay involved.

Selective-Reject ARQ

Window size limitation is more restrictive for selective-reject than for goback-N.

Selective-Reject ARQ

Consider the case of a 3-bit sequence number.

Allow a window size of seven, and consider the following scenario:

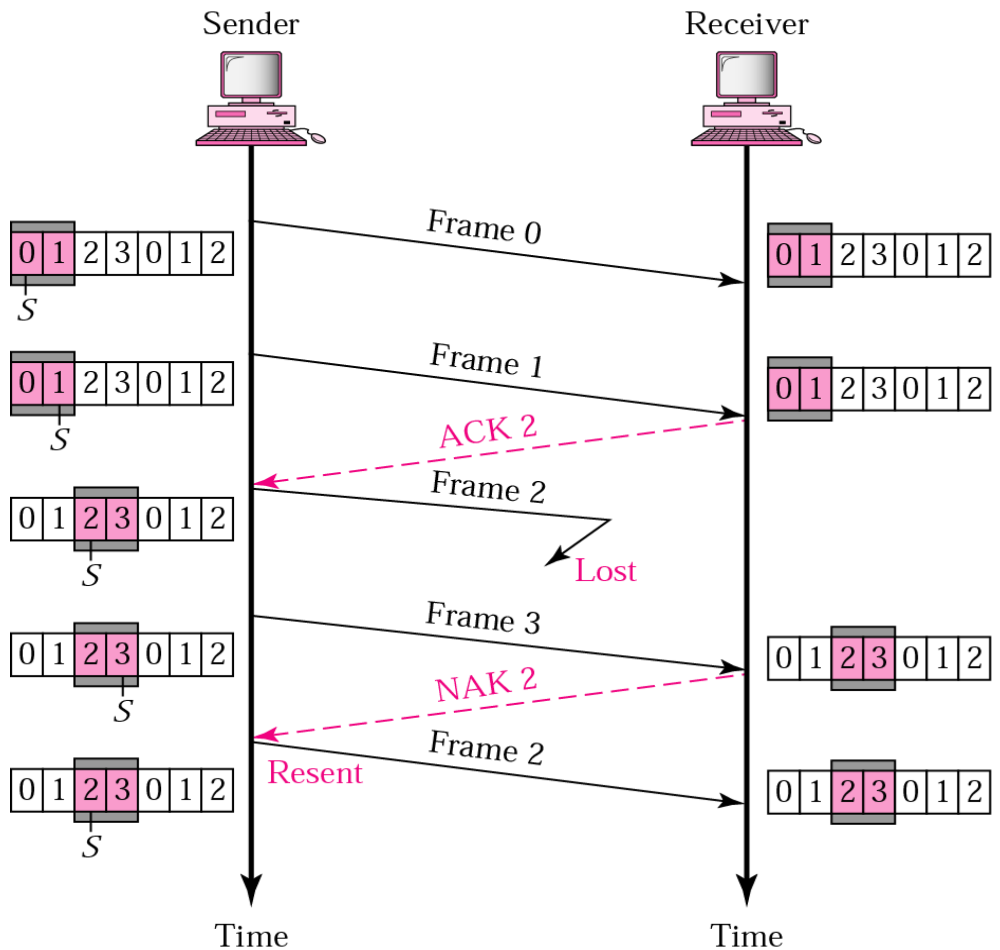
1. Station A sends frames 0 through 6 to station B.
2. Station B receives all seven frames and cumulatively acknowledges with RR 7.
3. Assume the RR 7 is lost.
4. A times out and retransmits frame 0.
5. B has already advanced its receive window to accept frames 7, 0, 1, 2, 3, 4, and 5.
   Thus it assumes that frame 7 has been lost and that this is a new frame 0, which it accepts.

Selective-Reject ARQ

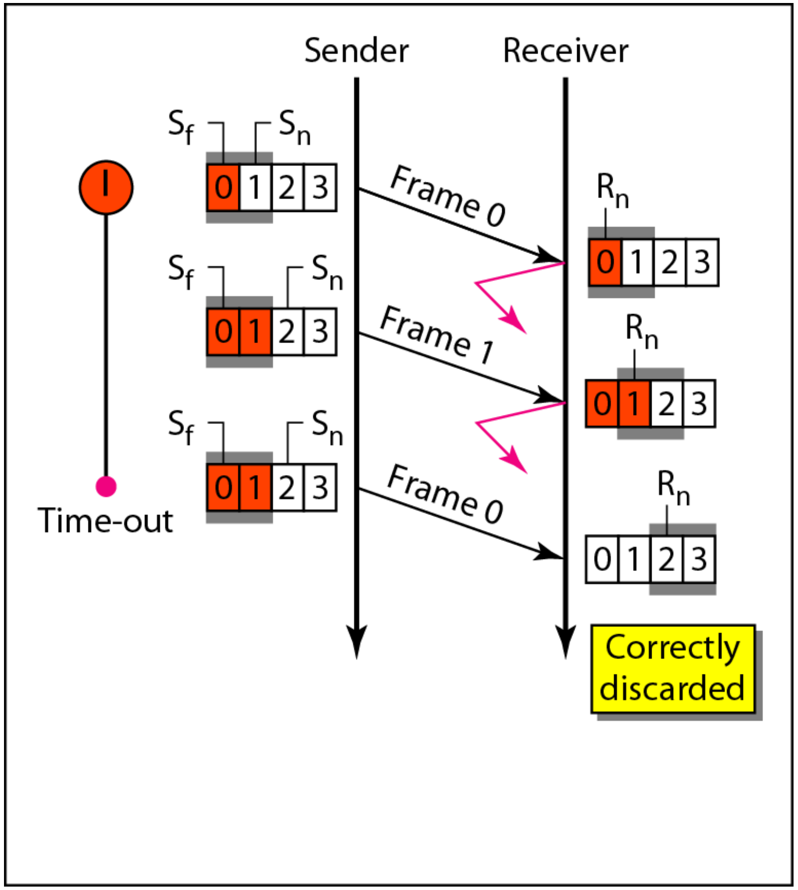
- To overcome the problem, the maximum window size should be no more than half the range of sequence numbers.

- In general, for a *k*-bit sequence number field, which provides a sequence number range of 2^k, the maximum window size is limited to 2^{k-1}.

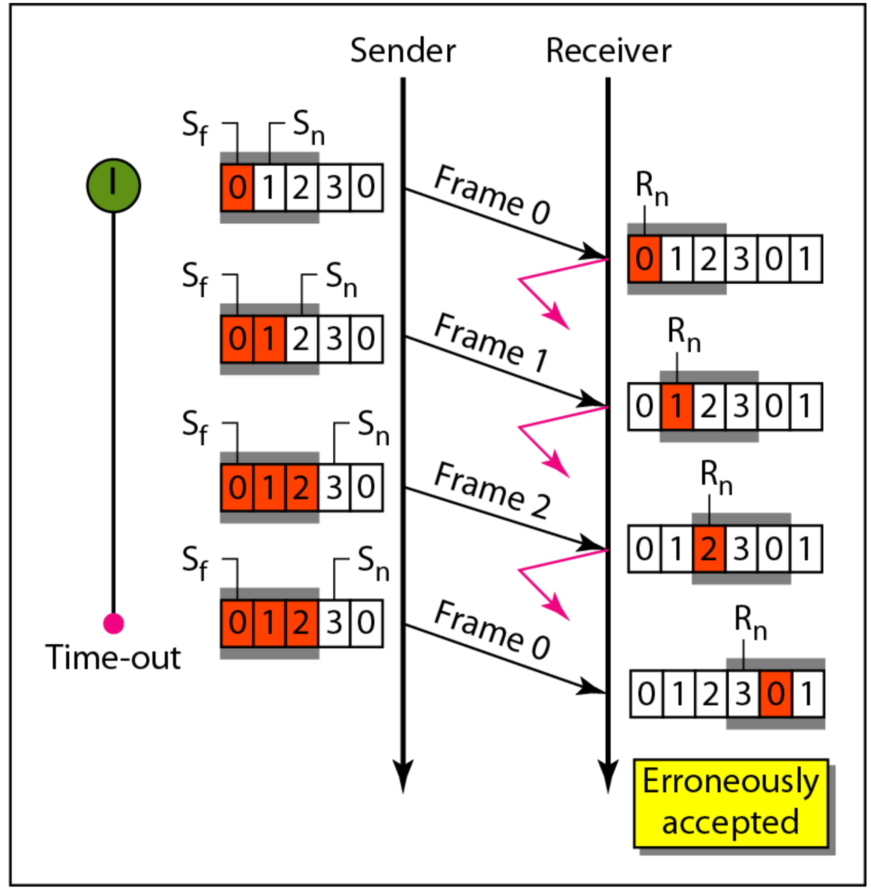
Selective Repeat ARQ, lost frame



Selective Repeat ARQ, window size



a. Window size = 2^{m-1}



b. Window size > 2^{m-1}

PERFORMANCE ISSUES

Stop-and-Wait Flow Control

Suppose that a long message is to be sent as a sequence of frames $F_1, F_2, \ldots\ldots F_n$ from S_1 to S_2

The total time to send the data, T, can be expressed as $T = nT_F$, where T_F is the time to send one frame and receive an acknowledgment. We can express T_F as follows:

$$T_F = t_{prop} + t_{frame} + t_{proc} + t_{prop} + t_{ack} + t_{proc}$$

where

t_{prop} = propagation time from S_1 to S_2

t_{frame} = time to transmit a frame (time for the transmitter to send out all of the bits of the frame)

t_{proc} = processing time at each station to react to an incoming event

t_{ack} = time to transmit an acknowledgment

PERFORMANCE ISSUES

Stop-and-Wait Flow Control

Assumptions:

- Processing time is relatively negligible
- Acknowledgment frame is very small compared to a data frame.

Then we can express the total time to send the data as

$$T_F = t_{\text{prop}} + t_{\text{frame}} + \boxed{t_{\text{proc}}} + t_{\text{prop}} + \boxed{t_{\text{ack}}} + \boxed{t_{\text{proc}}}$$

$$T = n(2t_{\text{prop}} + t_{\text{frame}})$$

PERFORMANCE ISSUES

Stop-and-Wait Flow Control

Of that time, only $n \times t_{\text{frame}}$ is actually spent transmitting data and the rest is overhead.

The utilization, or efficiency, of the line is

$$U = \frac{n \times t_{\text{frame}}}{n(2t_{\text{prop}} + t_{\text{frame}})} = \frac{t_{\text{frame}}}{2t_{\text{prop}} + t_{\text{frame}}}$$

Divide numerator and denominator by t_{frame} and substitute $a = t_{\text{prop}}/t_{\text{frame}}$

$$\boxed{U = \frac{1}{1 + 2a}}$$

PERFORMANCE ISSUES

Stop-and-Wait Flow Control

$$a = \frac{\text{Propagation Time}}{\text{Transmission Time}}$$

propagation time : distance *d* of the link divided by the velocity of propagation *V.*

propagation time = d/V

For unguided transmission through air or $V = 3 \times 10^8$ m/s.

For guided transmission, *V* is approximately 0.67 times the speed of light for optical fiber and copper media.

transmission time : length of the frame in bits, *L,* divided by the data rate *R.*

transmission time = L/R

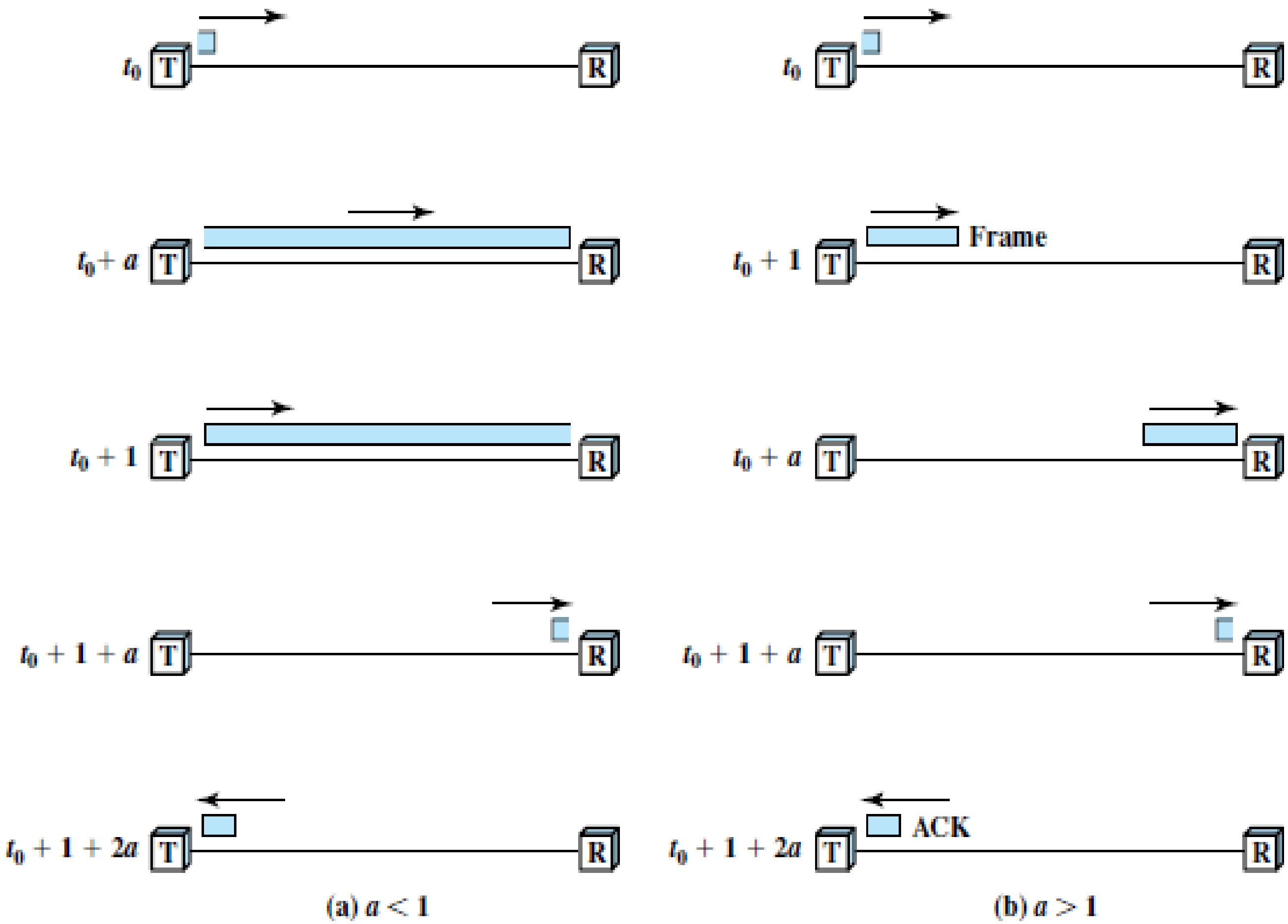


Figure 7.2 Stop-and-Wait Link Utilization (transmission time = 1; propagation time = a)

PERFORMANCE ISSUES

Error-Free Sliding-Window Flow Control

- To evaluate performance, we need to consider two cases:

  - **Case 1:** $W \geq 2a + 1$. The acknowledgment for frame 1 reaches A before A has exhausted its window. Thus, A can transmit continuously with no pause and normalized throughput is 1.0.
  - **Case 2:** $W < 2a + 1$. A exhausts its window at $t = W$ and cannot send additional frames until $t = 2a + 1$. Thus, normalized throughput is W time units out of a period of $(2a + 1)$ time units.

Therefore, express the utilization as

$$U = \begin{cases} 1 & W \geq 2a + 1 \\ \dfrac{W}{2a + 1} & W < 2a + 1 \end{cases}$$

Example 1

A system uses the Stop-and-Wait ARQ Protocol. If each frame carries 1000 bits of data, how long does it take to send 1 million bits of data if the distance between the sender and receiver is 5000 Km and the propagation speed is 2 x 10^8 m? Ignore transmission, waiting, and processing delays. Assume no data or control frame is lost or damaged.

Propagation delay=d/V
d=5000km=5 x 10^6 m, V= 2 x 10^8 m
Propagation delay=5 x 10^6 / 2 x 10^8 =25ms

Nr of frame=1000000/1000=1000
Time delay for one successful frame delivery is **50ms**(Ignore transmission, waiting, and processing delays. We assume no data or control frame is lost or damaged)
For **1000 frame**, time delay=1000x50ms=50s

Example 2

Consider the use of 1000-bit frames on a 1-Mbps satellite channel with a 270-ms delay. What is the maximum link utilization for

a) Stop-and-wait flow control?
b) Continuous flow control with a window size of 7?
c) Continuous flow control with a window size of 127?
d) Continuous flow control with a window size of 255?

a= propagation Time/(L/R)

$a=270 \times 10^{-3}/(10^{3}/ 10^{6}) = 270$

a)$U = 1/(1 + 2a) = 1/541 = 0.002$

b)$U = W/(1 + 2a) = 7/541 = 0.013$

c)$U = 127/541 = 0.23$

d)$U = 255/541 = 0.47$

Example 3

Using 5-bit sequence numbers, what is the maximum size of the send and receive windows for each of the following protocols?

a. Stop-and-Wait ARQ

b. Go-Back-NARQ

c. Selective-Repeat ARQ

a) Stop-and-Wait ARQ=2^0

b) Go-Back-NARQ=2^5 -1

c) Selective-Repeat ARQ=2^{5-1}

Example 4

A sender sends a series of frame to the same destination using 5-bit sequence numbers. If the sequence number starts with 0, what is the sequence number after sending 100 frame?

A five-bit sequence number can create sequence numbers from 0 to 31
The sequence number in the Nth frame is (N mod 32)
101th frame has the sequence number (101 mod 32) or **5**.

Example 5

A system uses the *Go-back-N* ARQ Protocol with a window size of 7. If each frame carries 1000 bits of data, how long does it take to send 1 million bits of data if the distance between the sender and receiver is 5000 Km and the propagation speed is 2×10^8 m?

Ignore the overhead due to the header and trailer. (Assume transmission time of frame=1ms)

We need to send w=7 frame

1000000/7000=143 windows

Transmission time of window =7x1ms=7ms

Delay for 1 window

$=t_{wframe}+2t_{prop}$ =7+50=57ms

Delay for 143 windows

=143x57ms=8.151s

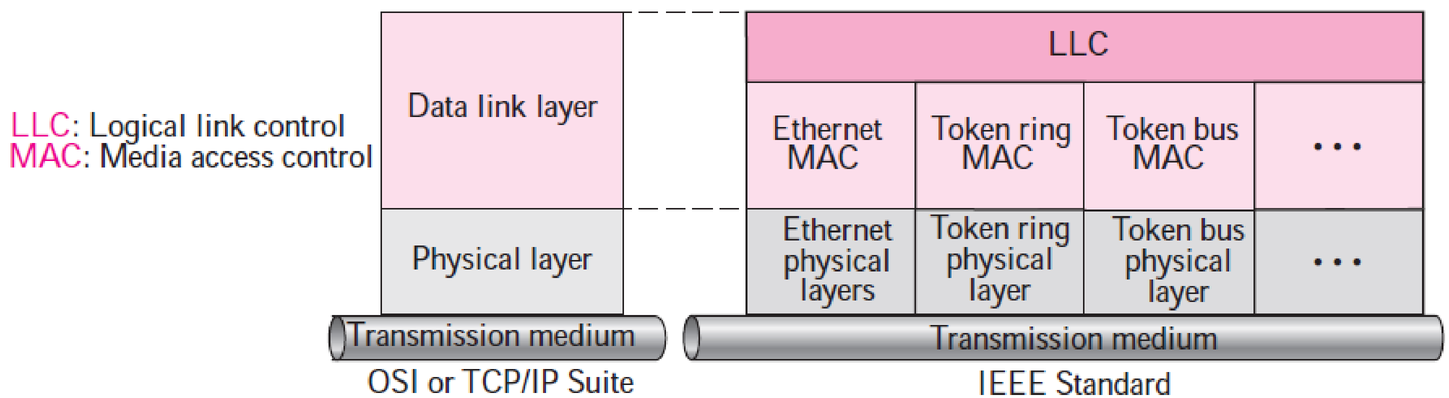
Ethernet Frame Format

IEEE Standards

In 1985, the Computer Society of the IEEE started a project, called **Project 802,** to set standards to enable intercommunication among equipment from a variety of manufacturers.

It is a way of specifying functions of the physical layer and the data link layer of major LAN protocols.

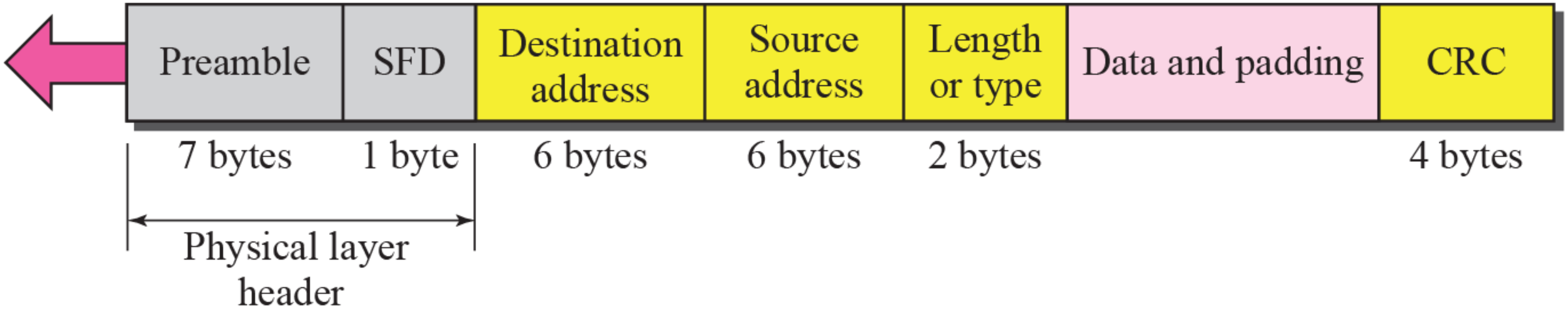
Figure 3.1 *IEEE standard for LANs*

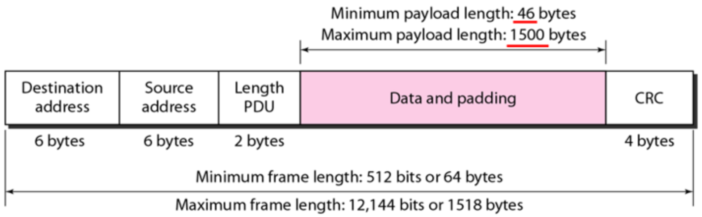


Ethernet Frame Format

Preamble: 56 bits of alternating 1s and 0s.

SFD: Start frame delimiter, flag (10101011)





Ethernet Frame Format

Preamble (7 bytes): 7 bytes (56 bits) of alternating 0s and 1s used to alert & synchronize receiver, sender clock rates.

SFD (1 byte): Start Frame Delimiter, flag (10101011): Signals the beginning of the frame. Last two bits 11 alert the receiver that the next field is the destination address.

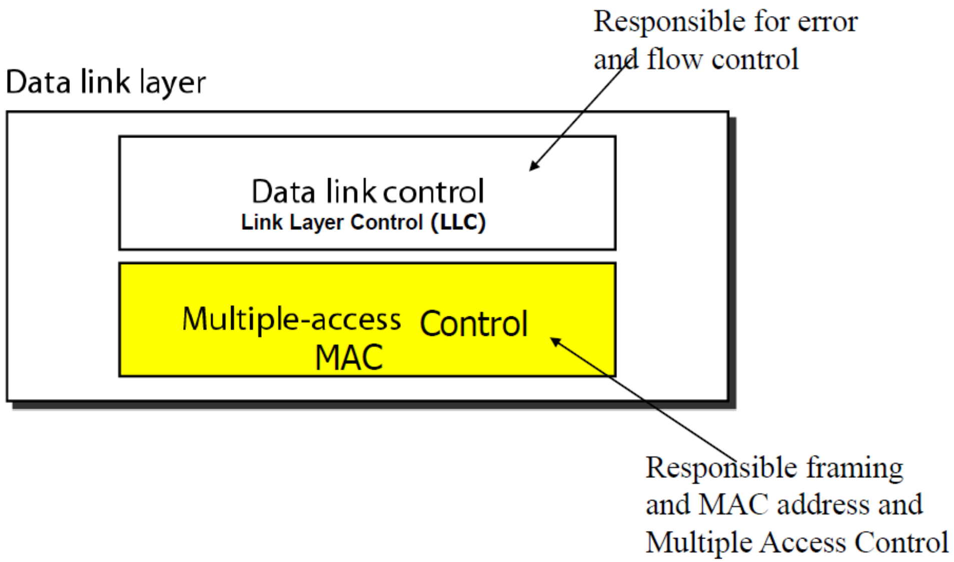
Destination and source address: MAC address of source and destination.

Type/Length (2 Bytes): Length: Up to 1500 (max length)
Type: Identify higher-layer data(0x0800 –IPv4 , 0x0806-ARP)

CRC(Cyclic Redundancy Check 4 Bytes): checked at receiver, if error is detected, the frame is simply dropped

Data: 46 to 1500 Bytes (if shorter: add pad)Why?

Access Control

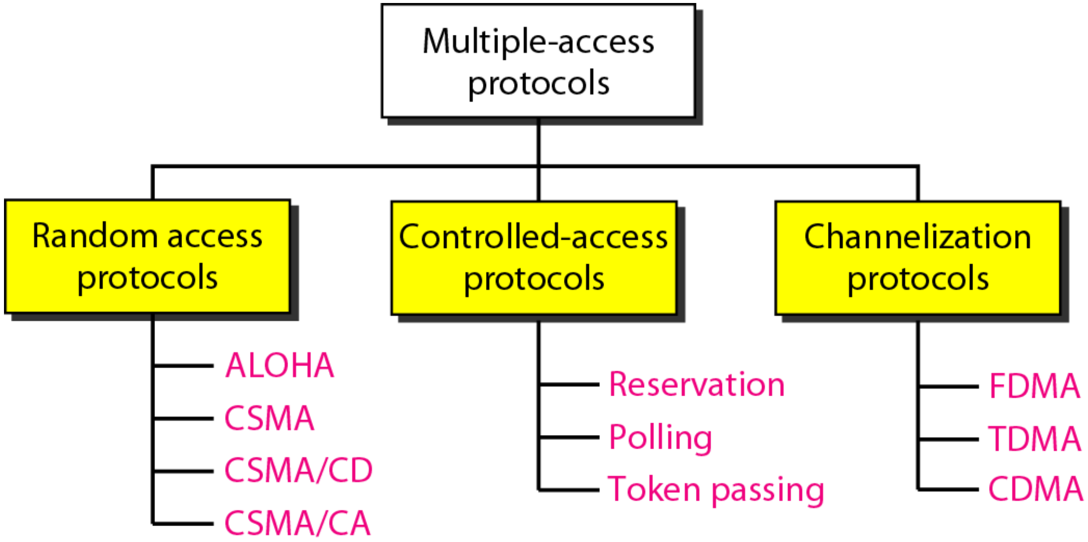


When two or more nodes transmit at the same time, their frames will collide and the link bandwidth is **wasted** during collision.

MAC protocol : to coordinate the transmission of the active nodes.

Multiple Access Protocols: Main task is to **minimize collisions** in order to **utilize the bandwidth** by:

- ✓ Determining **when** a station can use the link (medium)
- ✓ **what** a station should do when the link is **busy**
- ✓ **what** the station should do when it is involved in **collision**



CSMA/CD

The IEEE 802.3 standard defines **carrier sense multiple access with collision detection (CSMA/CD)** as the access method for traditional Ethernet.

To minimize the chance of collision and, the CSMA method was developed. The chance of collision can be reduced if a station senses the medium before trying to use it.

Carrier sense multiple access (CSMA) requires that each station first listen to the medium before sending.

CSMA can reduce the possibility of collision, but it cannot eliminate it.

CSMA/CD

The possibility of collision still exists because of propagation delay; when a station sends a frame, it still takes time (although very short) for the first bit to reach every station and for every station to sense it.

In other words, a station may sense the medium and find it idle, only because the first bit sent by another station has not yet been received.

CSMA/CD

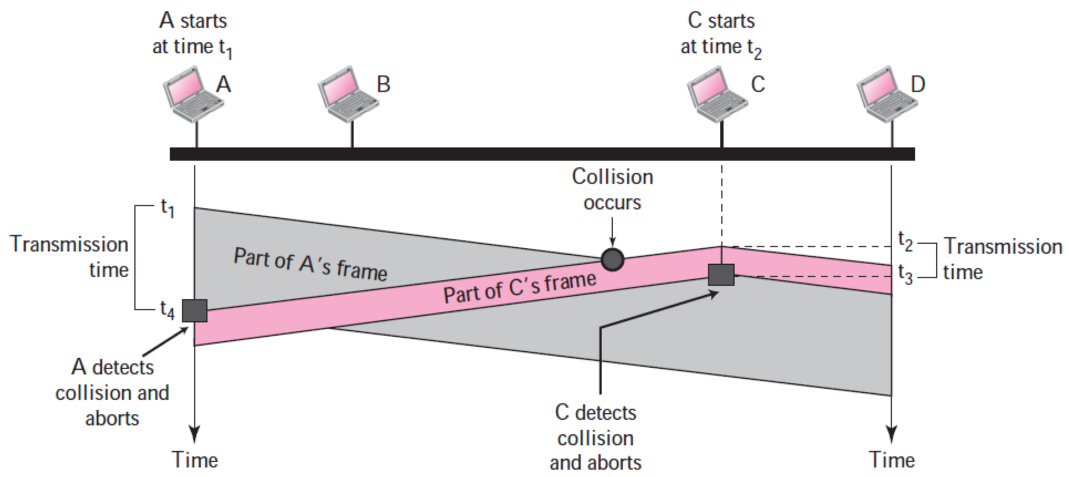
Carrier sense multiple access with collision detection (CSMA/CD) augments the algorithm to handle the collision.

In this method, a station monitors the medium after it sends a frame to see if the transmission was successful.

If so, the station data transmission is finished. If, however, there is a collision, the frame is sent again.

In Figure, stations A and C are involved in the collision.

Figure 3.8 *Collision of the first bit in CSMA/CD*

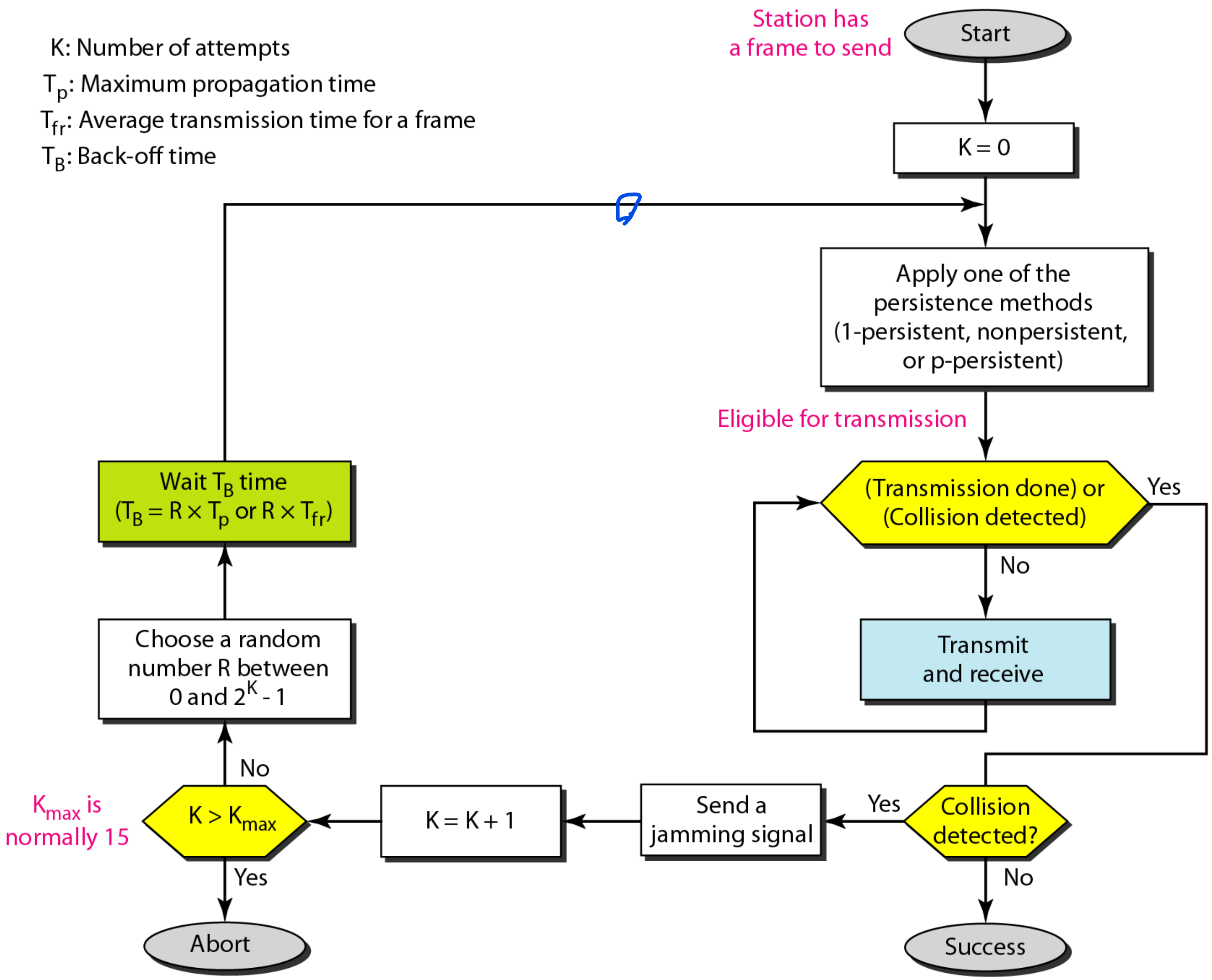


At time t_1, station A has started sending the bits of its frame. At time t_2, station C has not yet sensed the first bit sent by A. Station C starts sending the bits in its frame, which propagate both to the left and to the right.

The collision occurs sometime after time t_2. Station C detects a collision at time t_3 when it receives the first bit of A's frame. Station C immediately aborts transmission.

Station A detects collision at time t_4 when it receives the first bit of C's frame; it also immediately aborts transmission.

A transmits for the duration $t_4 - t_1$; C transmits for the duration $t_3 - t_2$.


K: Number of attempts
T_p: Maximum propagation time
T_{fr}: Average transmission time for a frame
T_B: Back-off time
Station has a frame to send
Start
K = 0
Apply one of the persistence methods (1-persistent, nonpersistent, or p-persistent)
Eligible for transmission
(Transmission done) or (Collision detected)
Yes
No
Transmit and receive
Collision detected?
Yes
No
Success
Send a jamming signal
K = K + 1
$K > K_{max}$
K_{max} is normally 15
No
Yes
Abort
Choose a random number R between 0 and $2^K - 1$
Wait T_B time ($T_B = R \times T_p$ or $R \times T_{fr}$)

Energy level during transmission, idleness, or collision

- level of energy in a channel can be: zero, normal, and abnormal.
- At the zero level, the channel is idle.
- Normal level : station has successfully captured the channel and is sending its frame.
- abnormal level : there is a collision and the level of the energy is twice the normal level.
- A station that has a frame to send or is sending a frame needs to monitor the energy level to determine if the channel is idle, busy, or in collision mode.