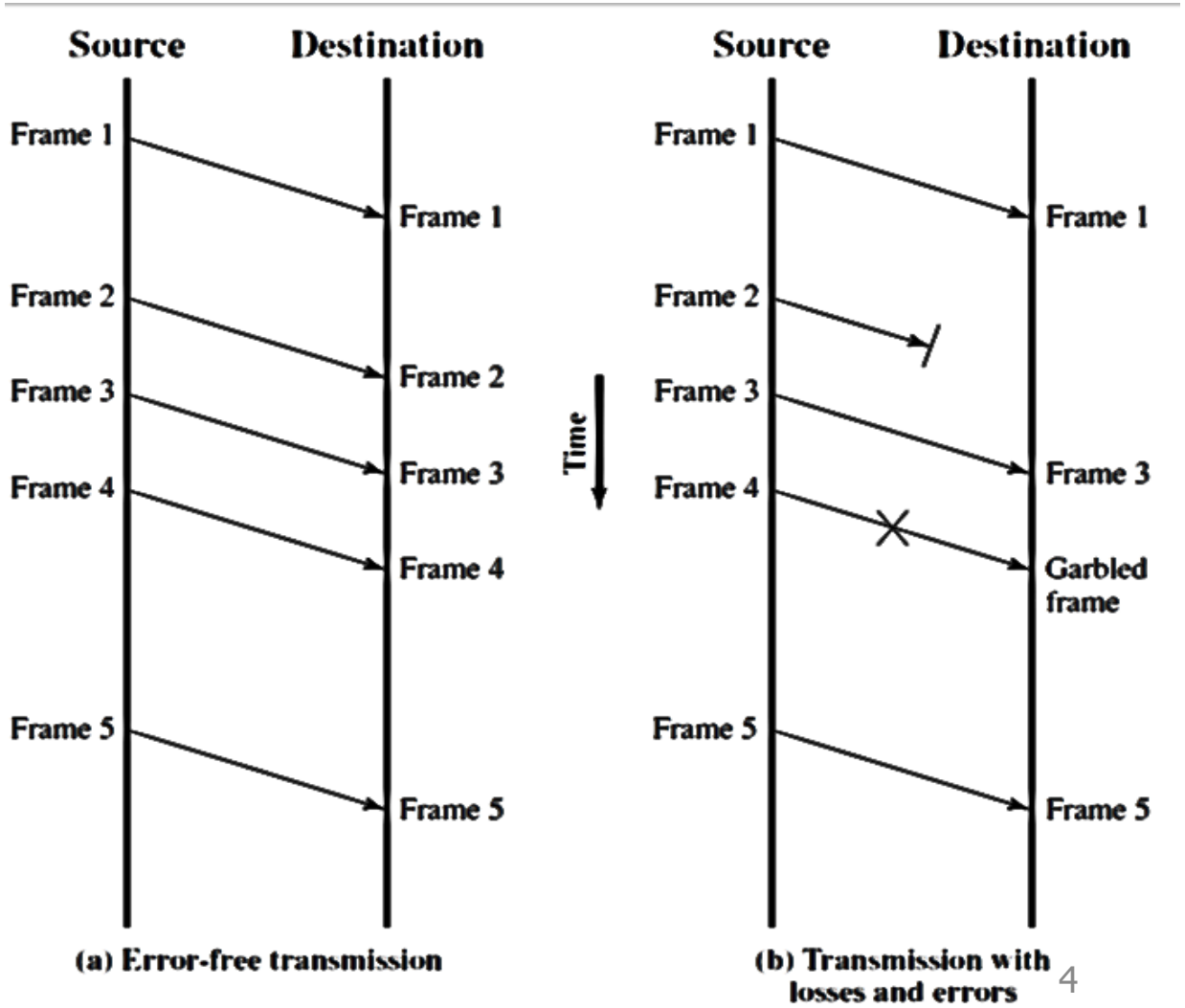
Feedback Error Control

- Error control refers to mechanisms to detect and correct errors that occur in the transmission of frames.
- Data are sent as a sequence of frames; frames arrive in the same order in which they are sent. There possibility of two types of errors:
- Lost frame: A frame fails to arrive at the other side. For example, a noise burst may damage a frame to the extent that the receiver is not aware that a frame has been transmitted.
- Damaged frame: A recognizable frame does arrive, but some of the bits are in error (have been altered during transmission).



(a) Error-free transmission

(b) Transmission with losses and errors

Feedback Error Control

- The most common techniques for error control are based on some or all of the following ingredients:
  - Error detection: As discussed earlier
  - Positive acknowledgment: The destination returns a positive acknowledgment to successfully received, error free frames.
  - Retransmission after timeout: The source retransmits a frame that has not been acknowledged after a predetermined amount of time.
  - Negative acknowledgment and retransmission: The destination returns a negative acknowledgment to frames in which an error is detected. The source retransmits such frames. This procedure is known as feedback error control.
- The process of retransmitting the data has traditionally been known as Automatic Repeat Request (ARQ). There are three types of ARQ that have been used: namely,
  - Stop and Wait,
  - Go back N
  - Selective Repeat.

ARQ Protocols

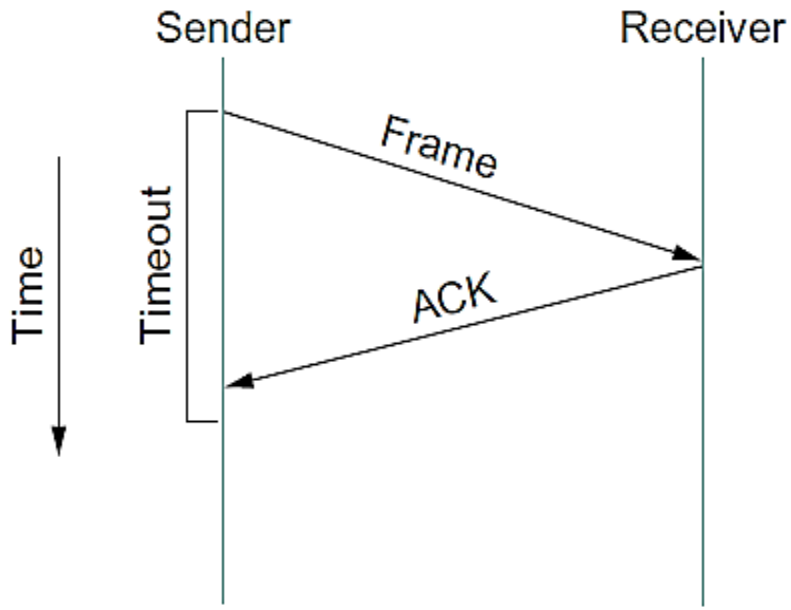
- Some important points:
  - Transmitter must buffer packets for possible retransmission (here packet refers to the packet sent by error control protocol, includes the message data and header information).
  - Feedback channel needs bandwidth as well.
  - Even for very few bit errors whole packet is retransmitted.
- ARQ protocols differ:
  - in the number of allowed outstanding frames /unacknowledged frames
  - in the buffering requirements at receiver / transmitter
  - in the way feedback is provided (positive / negative acknowledgement frames, timers)
- ARQ protocols also use acknowledgement packets, called ACKs, which is a small control frame that a protocol sends back to its peer(transmitter) saying that it has received an earlier frame.
- A control frame is a header without any data. The receipt of an acknowledgment indicates to the sender of the original frame that its frame was successfully delivered.
- If the sender does not receive an acknowledgment after a reasonable amount of time, then it retransmits the original frame. This action of waiting a reasonable amount of time is called a timeout.

Stop and Wait ARQ

- The simplest ARQ scheme is the stop and wait algorithm.
- The idea of stop and wait is straightforward: After transmitting one frame, the sender waits for an acknowledgment before transmitting the next frame.
- If the acknowledgment does not arrive after a certain period of time, the sender times out and retransmits the original frame.
- At the receiver, the data is checked for errors and if it is error free an acknowledgement (ACK) is sent back to the transmitter.
- If errors are detected at the receiver a negative acknowledgement (NAK) is returned.
- Since errors could equally occur in the ACK or NAK signals, they should also be checked for errors.
- Thus, only if each frame is received error free and an ACK is returned error free can the next frame be transmitted.
- Stop and wait ARQ guarantees in sequence delivery and is simple to implement.
- It requires one buffer at transmitter and one buffer at receiver.

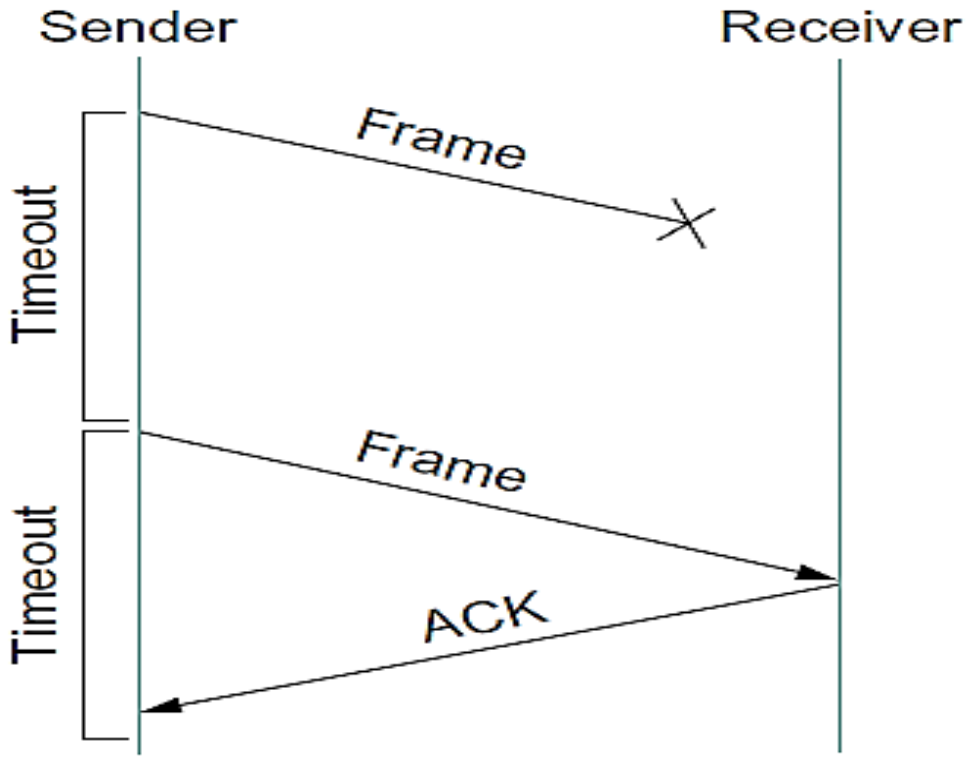
Stop and Wait ARQ

- This figure is a timeline, which is a common way to depict a protocol's behavior.
- The sending side is represented on the left, the receiving side is depicted on the right, and time flows from top to bottom.
- Figure shows the situation in which the ACK is received before the timer expires.



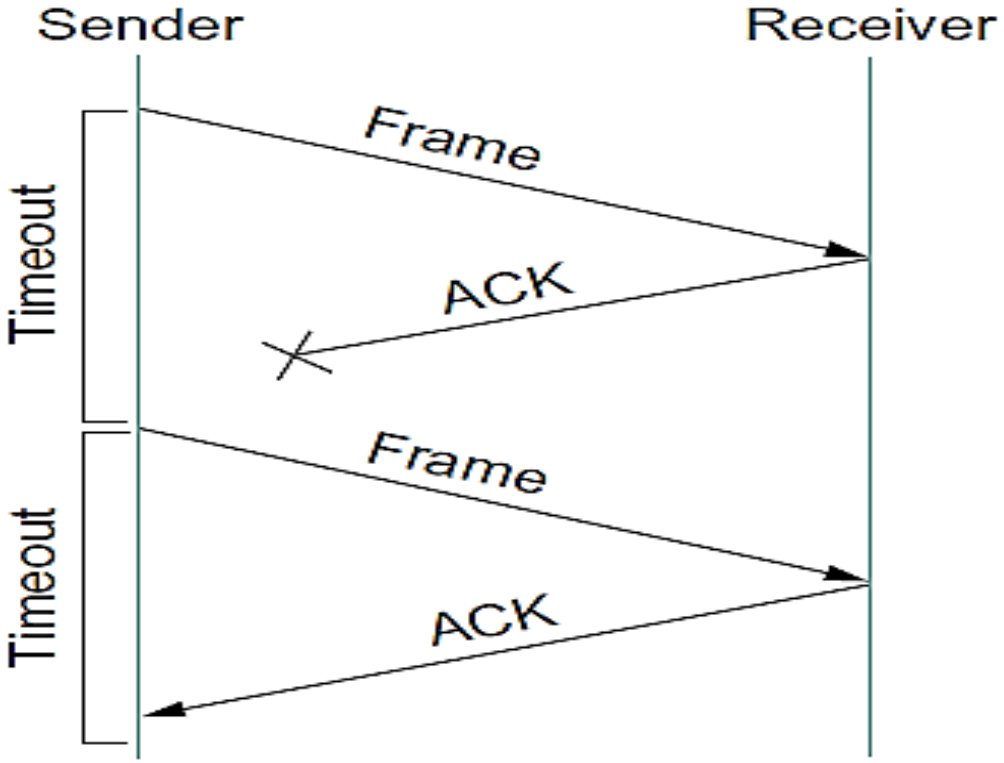
Stop and Wait ARQ

- This figure is a timeline, which is a common way to depict a protocol's behavior.
- The sending side is represented on the left, the receiving side is depicted on the right, and time flows from top to bottom.
- Figure shows the situation in which the original frame is lost.
- By "lost" we mean that the frame was corrupted while in transit, that this corruption was detected by an error code on the receiver, and that the frame was subsequently discarded.



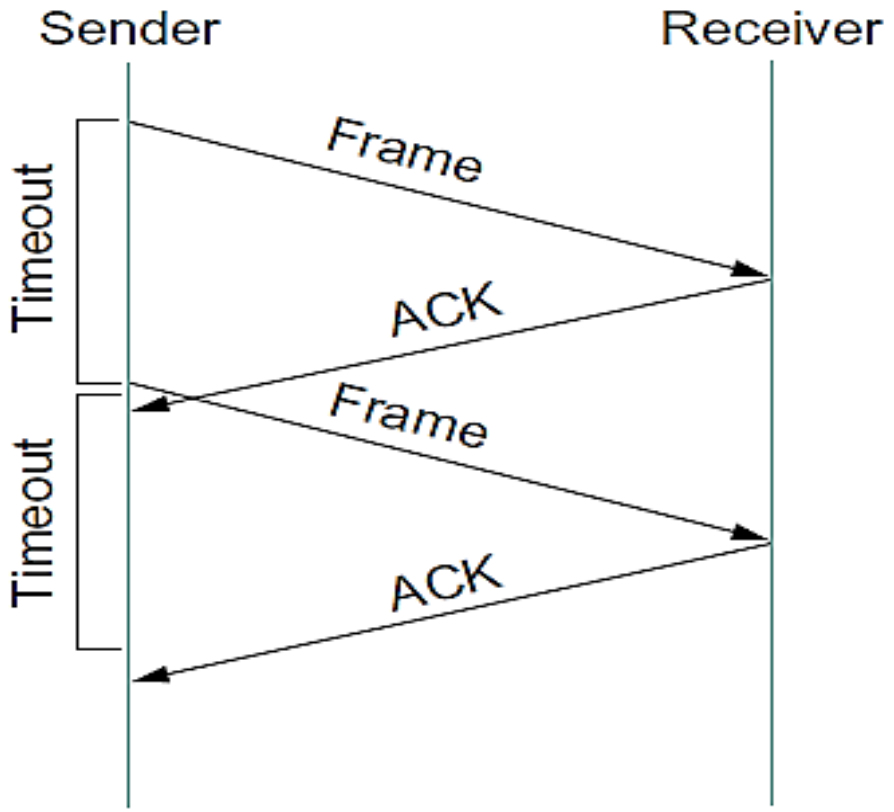
Stop and Wait ARQ

- This figure is a timeline, which is a common way to depict a protocol's behavior.
- The sending side is represented on the left, the receiving side is depicted on the right, and time flows from top to bottom.
- Figure shows the situation in which the ACK is lost.



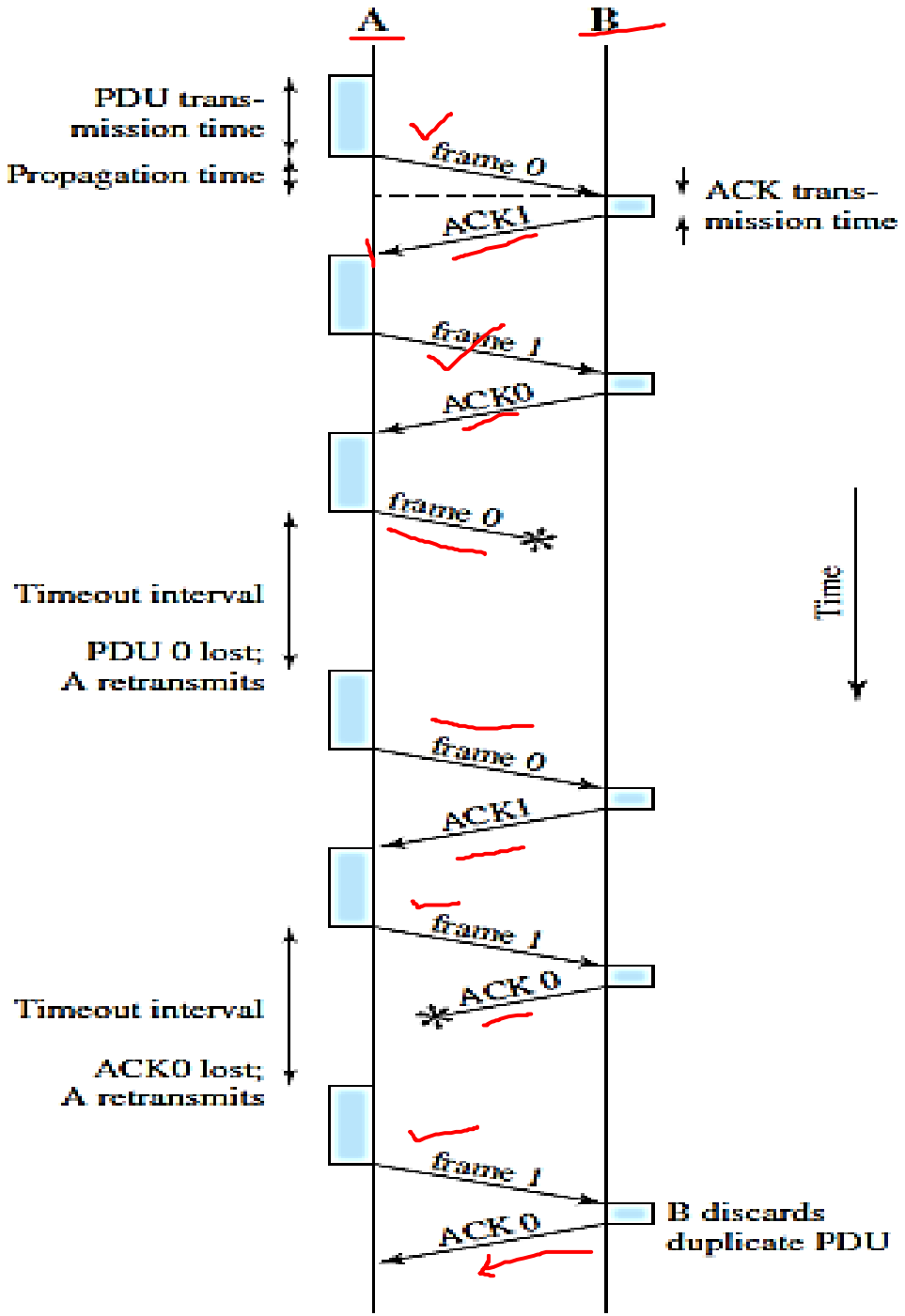
Stop and Wait ARQ

- This figure is a timeline, which is a common way to depict a protocol's behavior.
- The sending side is represented on the left, the receiving side is depicted on the right, and time flows from top to bottom.
- Figure shows the situation in which the timeout fires too soon.



Stop and Wait ARQ

- Suppose the sender sends a frame and the receiver acknowledges it, but the acknowledgment is either lost or delayed in arriving.
- This situation is illustrated in timelines on slide # 10 and 11.
- In both cases, the sender times out and retransmits the original frame, but the receiver will think that it is the next frame, since it correctly received and acknowledged the first frame.
- This has the potential to cause duplicate copies of a frame to be delivered.
- To address this problem, the header for a stop and wait protocol usually includes a 1 bit sequence number - that is, the sequence number can take on the values 0 and 1 - and the sequence numbers used for each frame alternate.
- Thus, when the sender retransmits frame 0, the receiver can determine that it is seeing a second copy of frame 0 rather than the first copy of frame 1 and therefore can ignore it (the receiver still acknowledges it, in case the first ACK was lost).


A
B
PDU trans-
mission time
Propagation time
frame 0
ACK trans-
mission time
ACK1
frame 1
ACK0
frame 0
Timeout interval
PDU 0 lost;
A retransmits
Time
frame 0
ACK1
frame 1
ACK 0
Timeout interval
ACK0 lost;
A retransmits
frame 1
B discards
duplicate PDU
ACK 0

Stop and Wait ARQ

- The main shortcoming of the stop and wait algorithm is that it allows the sender to have only one outstanding frame on the link at a time, and this may be far below the link's capacity.
- Consider, for example, a 1.5 Mbps link with a 45 ms round trip time.
- This link has a:

$$delay \times bandwidth = 1.5\,Mbps * 45ms = 8\,KB.$$

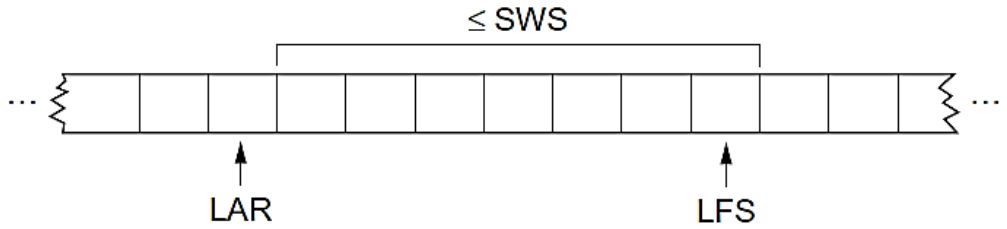
The significance of the delay × bandwidth product is that it represents the amount of data that could be in transit.

- Since the sender can send only one frame per RTT, and assuming a frame size of 1 KB, about one eighth of the link's capacity is used. To use the link fully, then, we'd like the sender to be able to transmit up to eight frames before having to wait for an acknowledgment.

Sliding Window ARQ

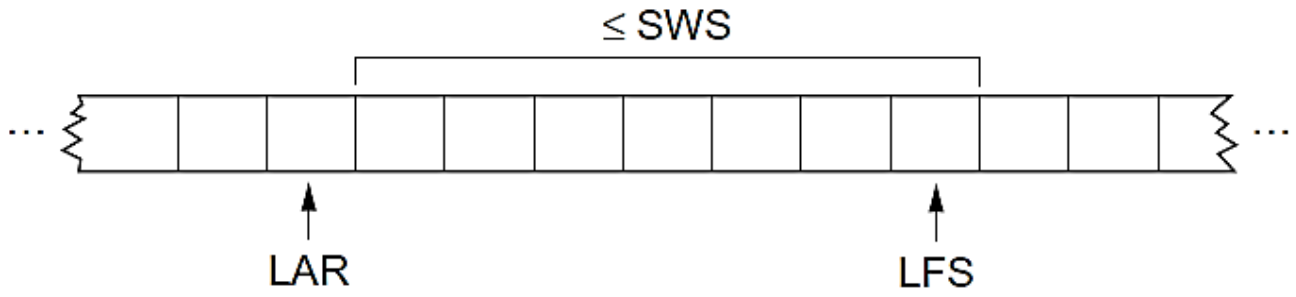
- The sliding window algorithm works as follows.
- First, the sender assigns a sequence number, denoted SeqNum, to each frame.
- The sender maintains three variables:
  - The send window size, denoted SWS, gives the upper bound on the number of outstanding (unacknowledged) frames that the sender can transmit;
  - LAR denotes the sequence number of the last acknowledgment received;
  - and LFS denotes the sequence number of the last frame sent.
- The sender also maintains the following invariant:

$$LFS - LAR \leq SWS$$



Sliding Window ARQ

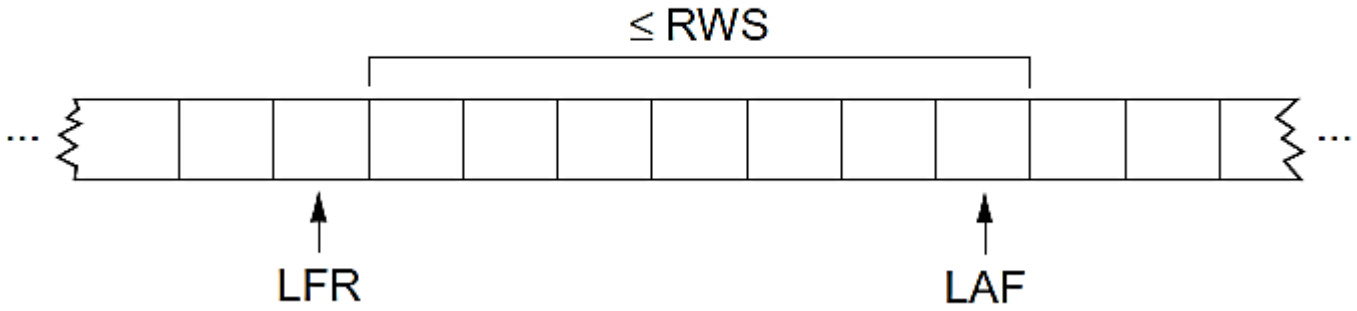
- When an acknowledgment arrives, the sender moves LAR to the right, thereby allowing the sender to transmit another frame.
- Also, the sender associates a timer with each frame it transmits, and it retransmits the frame if the timer expire before an ACK is received.
- Sender has to buffer up to SWS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



Sliding Window ARQ

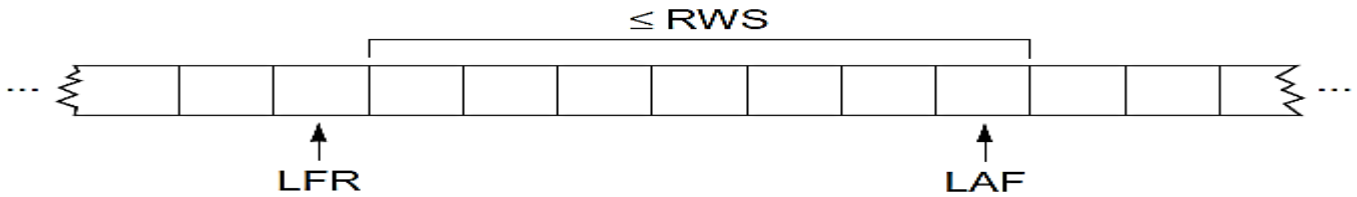
- The receiver maintains the following three variables:
  - The receive window size, denoted RWS, gives the upper bound on the number of out of order frames that the receiver is willing to accept;
  - LAF denotes the sequence number of the largest acceptable frame;
  - LFR denotes the sequence number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

$$LAF - LFR \leq RWS$$



Sliding Window ARQ

- When a frame with sequence number SeqNum arrives, the receiver takes the following action.
- If $SeqNum \leq LFR \; or \; SeqNum > LAF$, then the frame is outside the receiver's window and it is discarded.
- If $LFR < SeqNum \leq LAF$, then the frame is within the receiver's window and it is accepted.
- 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 $LAF = LFR + RWS$.



Go Back N ARQ

- Go back N combats the inefficiency of Stop and Wait by allowing N outstanding frames where N is also called window size and outstanding are frames not yet acknowledged.
- While no errors occur, the destination will acknowledge incoming frames as usual (RR = receive ready).
- If the destination station detects an error in a frame, it may send a negative acknowledgment (REJ = reject)for that frame.
- The destination station will discard that frame and all future incoming frames until the frame in error is correctly received.
- Thus, the 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

Scenario

- 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 and A has just transmitted frame i.
- The go back N technique takes into account the following steps:

1. Damaged frame**.** If the received frame is invalid (i.e., B detects an error, or the frame is so damaged that B does not even perceive that it has received a frame), B discards the frame and takes no further action as the result of that 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.

(b) 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

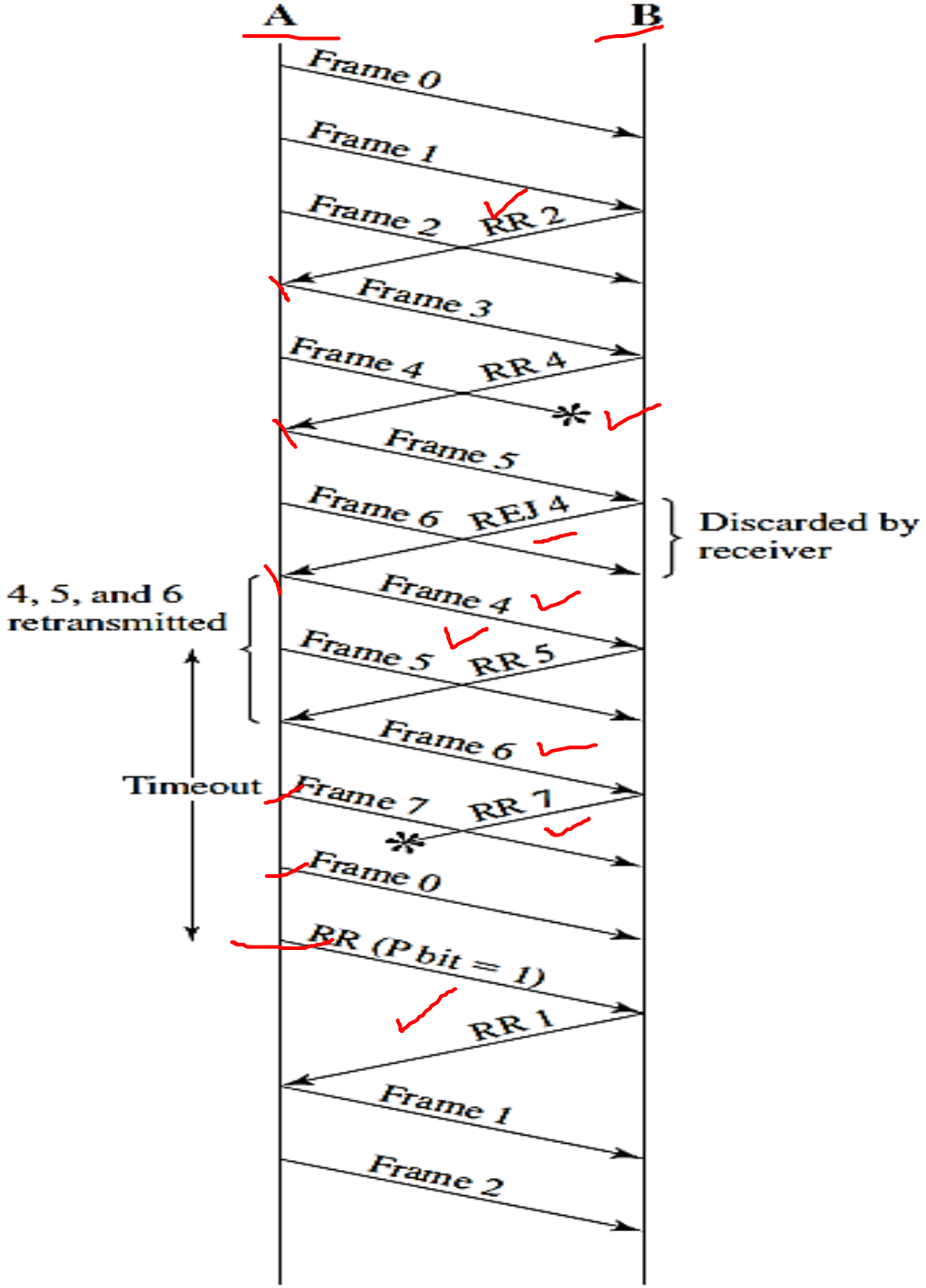
Scenario

2. Damaged RR. There are two subcases:

(a) B receives frame i and sends RR 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. 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 as discussed in previous slide.

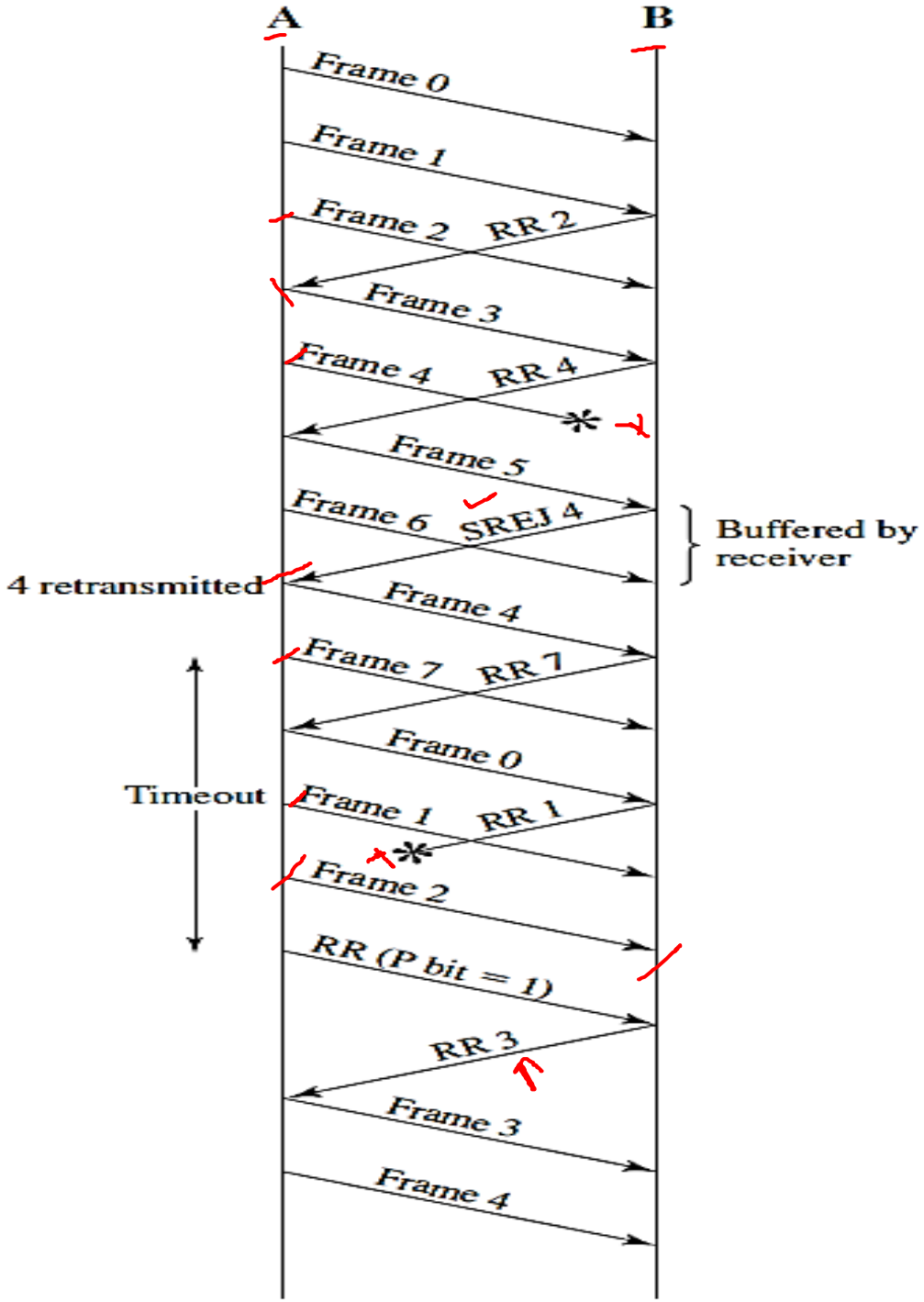

A
B
Frame 0
Frame 1
Frame 2
RR 2
Frame 3
Frame 4
RR 4
*
Frame 5
Frame 6
REJ 4
Discarded by receiver
4, 5, and 6 retransmitted
Frame 4
Frame 5
RR 5
Frame 6
Timeout
Frame 7
RR 7
*
Frame 0
RR (P bit = 1)
RR 1
Frame 1
Frame 2

Go Back N Properties

- The transmitter needs N buffers, the receiver only a single buffer to accept an incoming packet
- If a packet fails, this packet and all subsequent packets are retransmitted, even if the latter were correctly received.
- If the packet error rate (PER) is small, Go back N is reasonably efficient, but for higher PERs the protocol retransmits many correctly received packets and becomes inefficient.

Selective Repeat ARQ

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


A
B
Frame 0
Frame 1
Frame 2
RR 2
Frame 3
Frame 4
RR 4
*
Frame 5
Frame 6
SREJ 4
Buffered by receiver
4 retransmitted
Frame 4
Frame 7
RR 7
Frame 0
Timeout
Frame 1
RR 1
*
Frame 2
RR (P bit = 1)
RR 3
Frame 3
Frame 4

Flow Control

- Flow control is a technique for assuring that a transmitting entity does not overwhelm a receiving entity with data.
- The receiving entity typically allocates a data buffer of some maximum length for a transfer.
- When data are received, the receiver must do a certain amount of processing before passing the data to the higher layers.
- In the absence of flow control, the receiver's buffer may fill up and overflow while it is processing old data.
- For this section, we assume that all frames that are transmitted are successfully received; no frames are lost and none arrive with errors.
- Flow control Mechanism used:
  - Stop and wait Flow Control
  - Sliding Window Flow Control

Stop and Wait Flow Control

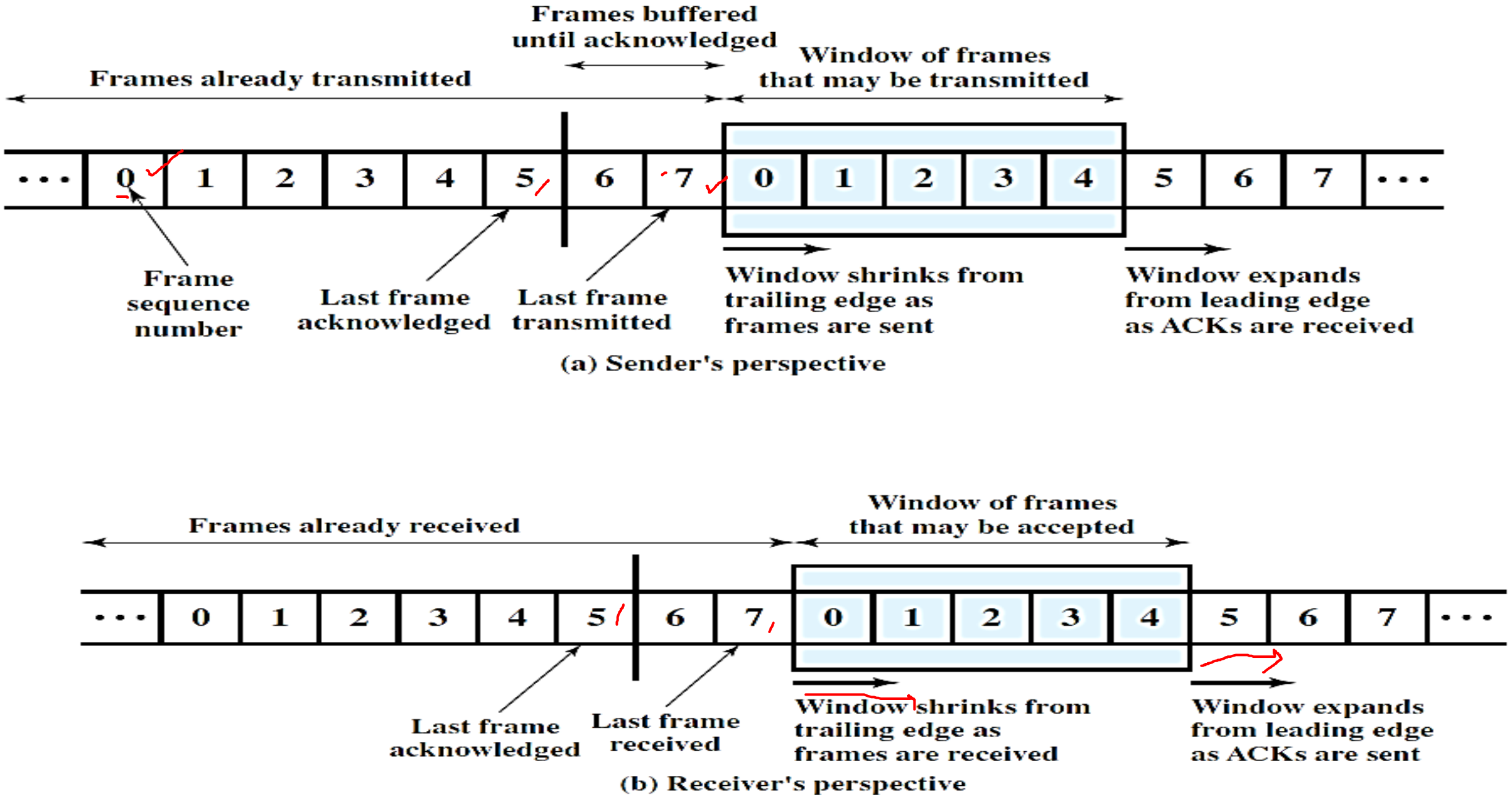
- The simplest form of flow control, known as stop and wait flow control.
- It works as follows:
  - A source entity transmits a frame. After the destination entity receives the frame, it indicates its willingness to accept another frame by sending back an acknowledgment to the frame just received.
  - 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 withholding acknowledgment.
- The main shortcoming of the stop and wait algorithm is that it allows the sender to have only one outstanding frame on the link at a time, and this may be far below the link's capacity.

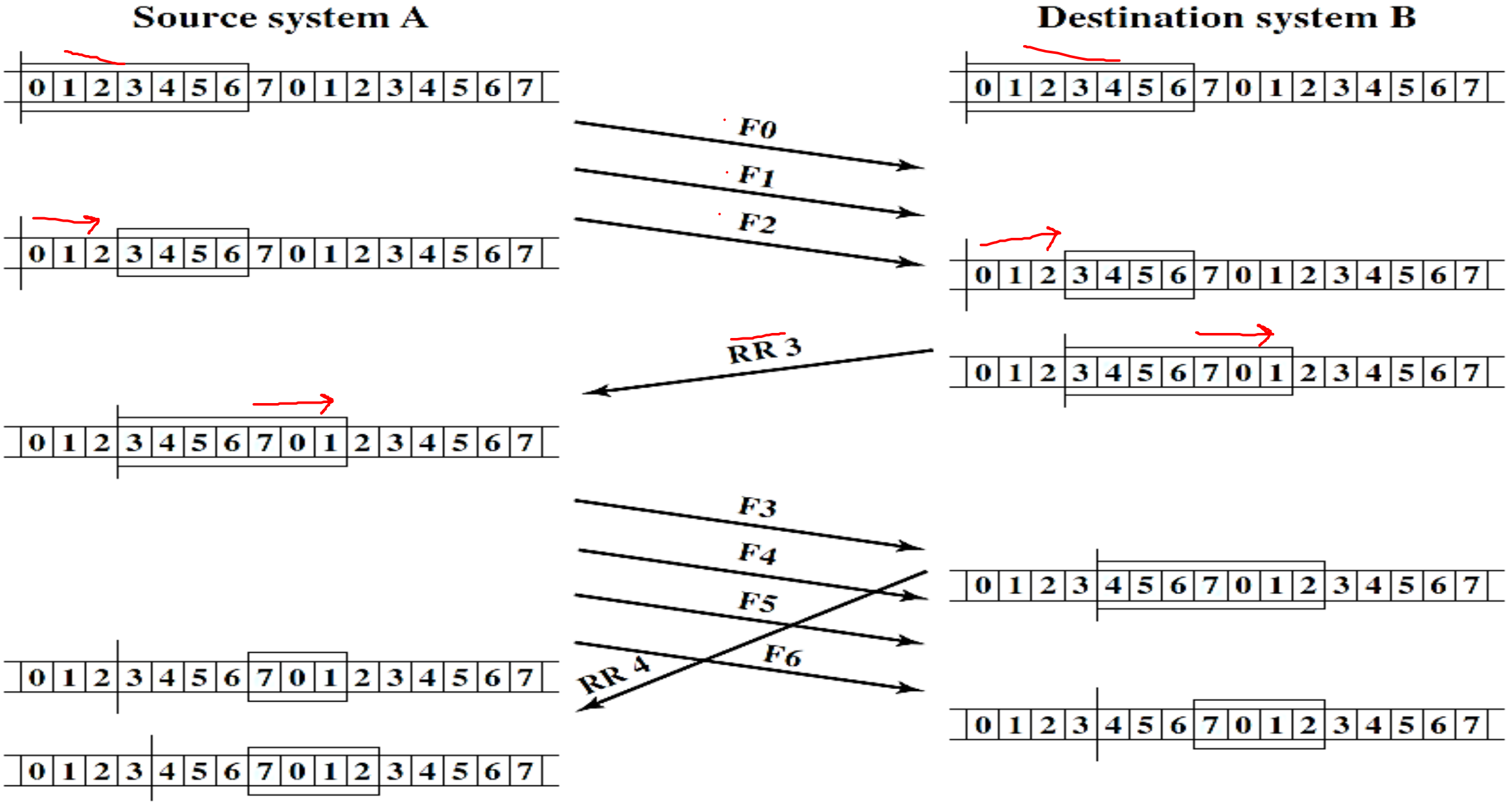
Sliding Window Flow Control

- The essence of the problem in stop and wait flow control is that only one frame at a time can be in transit.
- In situations where the bit length of the link is greater than the frame length serious inefficiencies result. Efficiency can be greatly improved by allowing multiple frames to be in transit at the same time.
- Let us assume two stations, A and B, connected via a full duplex link. 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.
- 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.
- This acknowledgment also implicitly announces that B is prepared to receive the next W frames, beginning with the number specified.
- This scheme can also be used to acknowledge multiple frames. For example, 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.
- A maintains a list of sequence numbers that it is allowed to send, and B maintains a list of sequence numbers that it is prepared to receive.
- 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

- Some more comments on sliding window flow control.
- Sequence number to be used occupies a field in the frame, therefore , it is limited to a range of values.
- For example, for a 3 bit field, the sequence number can range from 0 to 7. Accordingly, frames are numbered modulo 8; that is, after sequence number 7, the next number is 0.
- In general, for a k bit field the range of sequence numbers is 0 through 2^k-1 and frames are numbered modulo2^k.
- The maximum window size is $2^k - 1$.
- Most data link control protocols also allow a station to cut off the flow of frames from the other side by sending a Receive Not Ready (RNR) message, which acknowledges former frames but forbids transfer of future frames.
- Thus, RNR 5 means "I have received all frames up through number 4 but am unable to accept any more at this time." At some subsequent point, the station must send a normal acknowledgment to reopen the window.


Frames buffered
until acknowledged
Window of frames
that may be transmitted
Frames already transmitted
• • •
0
1
2
3
4
5
6
7
0
1
2
3
4
5
6
7
• • •
Frame
sequence
number
Last frame
acknowledged
Last frame
transmitted
Window shrinks from
trailing edge as
frames are sent
Window expands
from leading edge
as ACKs are received
(a) Sender's perspective
Window of frames
that may be accepted
Frames already received
• • •
0
1
2
3
4
5
6
7
0
1
2
3
4
5
6
7
• • •
Last frame
acknowledged
Last frame
received
Window shrinks from
trailing edge as
frames are received
Window expands
from leading edge
as ACKs are sent
(b) Receiver's perspective


Source system A
Destination system B
0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
F0
F1
F2
RR 3
F3
F4
F5
F6
RR 4

Piggybacking

- So far, transmission in one direction only. If two stations exchange data, each needs to maintain two windows, one for transmit and one for receive, and each side needs to send the data and acknowledgments to the other.
- 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.
- When a data frame arrives, instead of immediately sending a separate control frame, the receiver restrains itself and waits until the network layer passes it the next packet.
- The acknowledgement is attached to the outgoing data frame (using the ack field in the frame header).
- In effect, the acknowledgement gets a free ride on the next outgoing data frame.
- The technique of temporarily delaying outgoing acknowledgements so that they can be hooked onto the next outgoing data frame is known as piggybacking.

Piggybacking

- The principal advantage of using piggybacking over having distinct acknowledgement frames is a better use of the available channel bandwidth.
- The ack field in the frame header costs only a few bits, whereas a separate frame would need a header, the acknowledgement, and a checksum.
- Thus, if a station has data to send and an acknowledgment to send, it sends both together in one frame, saving communication capacity.
- 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 new 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.
- Sliding window flow control is potentially much more efficient than stop and wait flow control. The reason is that, with sliding window flow control, the transmission link is treated as a pipeline that may be filled with frames in transit. In contrast, with stop and wait flow control, only one frame may be in the pipe at a time.