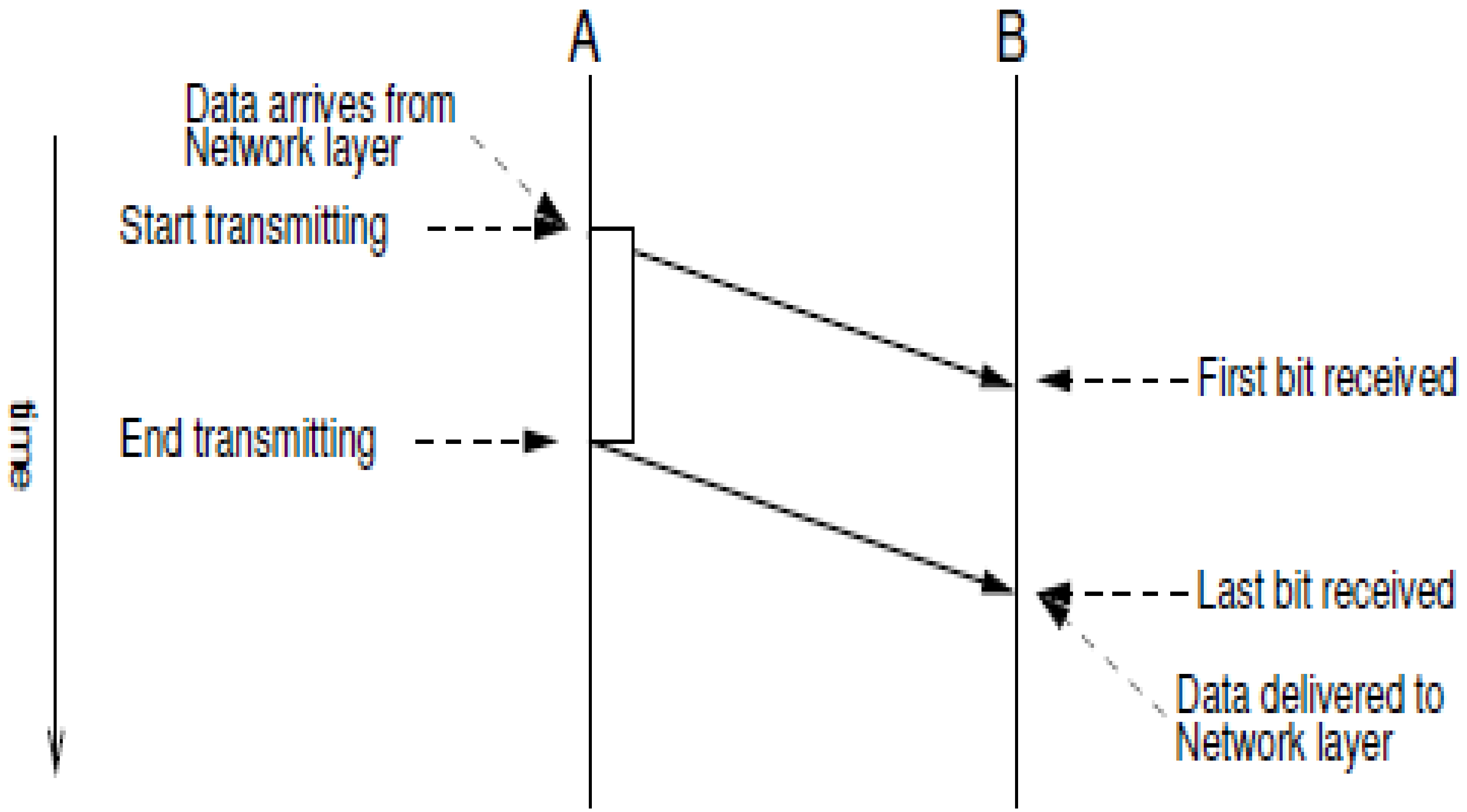
Flow Control
chap-7

Data Link Control Protocols

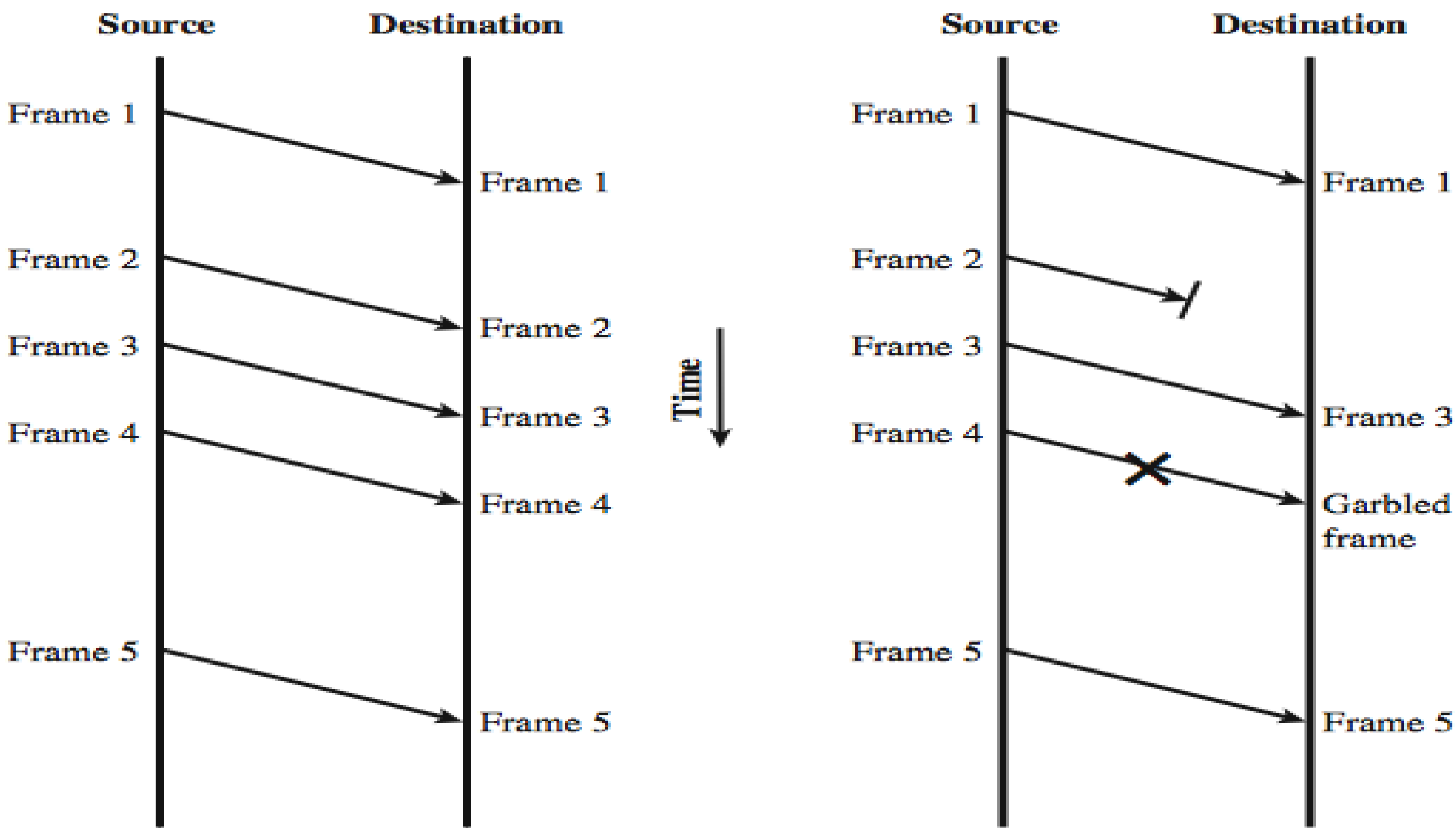
- Physical layer concentrates on sending signals over transmission link
- More control and management is needed to send data over data communications link
  - Frame synchronization: start and end of each frame
  - Flow control: ensure sender does not send too fast for receiver
  - Error control: correct bit errors introduced by transmission system
  - Addressing: must specify identity of two stations communicating
  - Control and data: receiver must distinguish between control and data information
  - Link management: setup and maintain the link
- We will focus on Flow Control and Error Control

Flow Control

- Flow control aims to ensure sending entity does not overwhelm receiving entity.
  - If sender sends too fast for receiver, then buffer may overflow
  - Result of buffer overflow: data is lost, possibly need to retransmit, which reduces performance
  - Flow control tries to prevent buffer overflow
  - Assume no errors but varying delays


A
B
Data arrives from
Network layer
Start transmitting
End transmitting
time
First bit received
Last bit received
Data delivered to
Network layer

Model of Frame Transmission



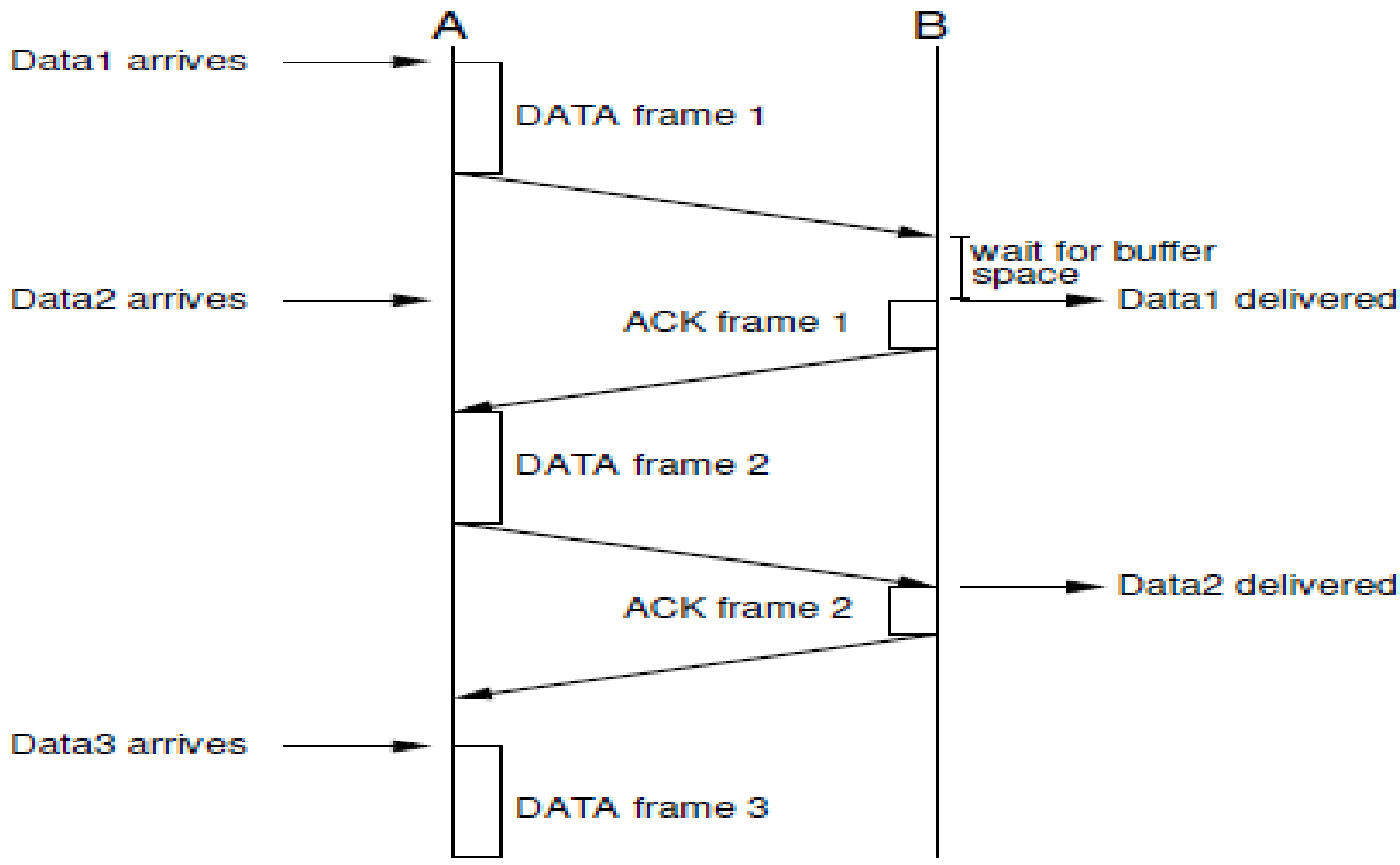
(a) Error-free transmission

(b) Transmission with losses and errors

Stop-and-Wait Flow Control

- Frame Types
  - DATA: contains information to be sent
  - ACKnowledgement: acknowledges receipt of data
- Rules
  - Source transmits a DATA frame
  - Source waits for ACK frame before sending next DATA frame
  - Destination receives DATA frame and replies with an ACK if ready for more data
  - Destination can stop flow of data by not sending ACK

Stop-and-Wait Flow Control



Stop and Wait Link Utilization

Propagation delay is the time it takes for a bit to travel from sender to receiver, depends on

- **transmission media and**
- **the distance between sender and receiver**

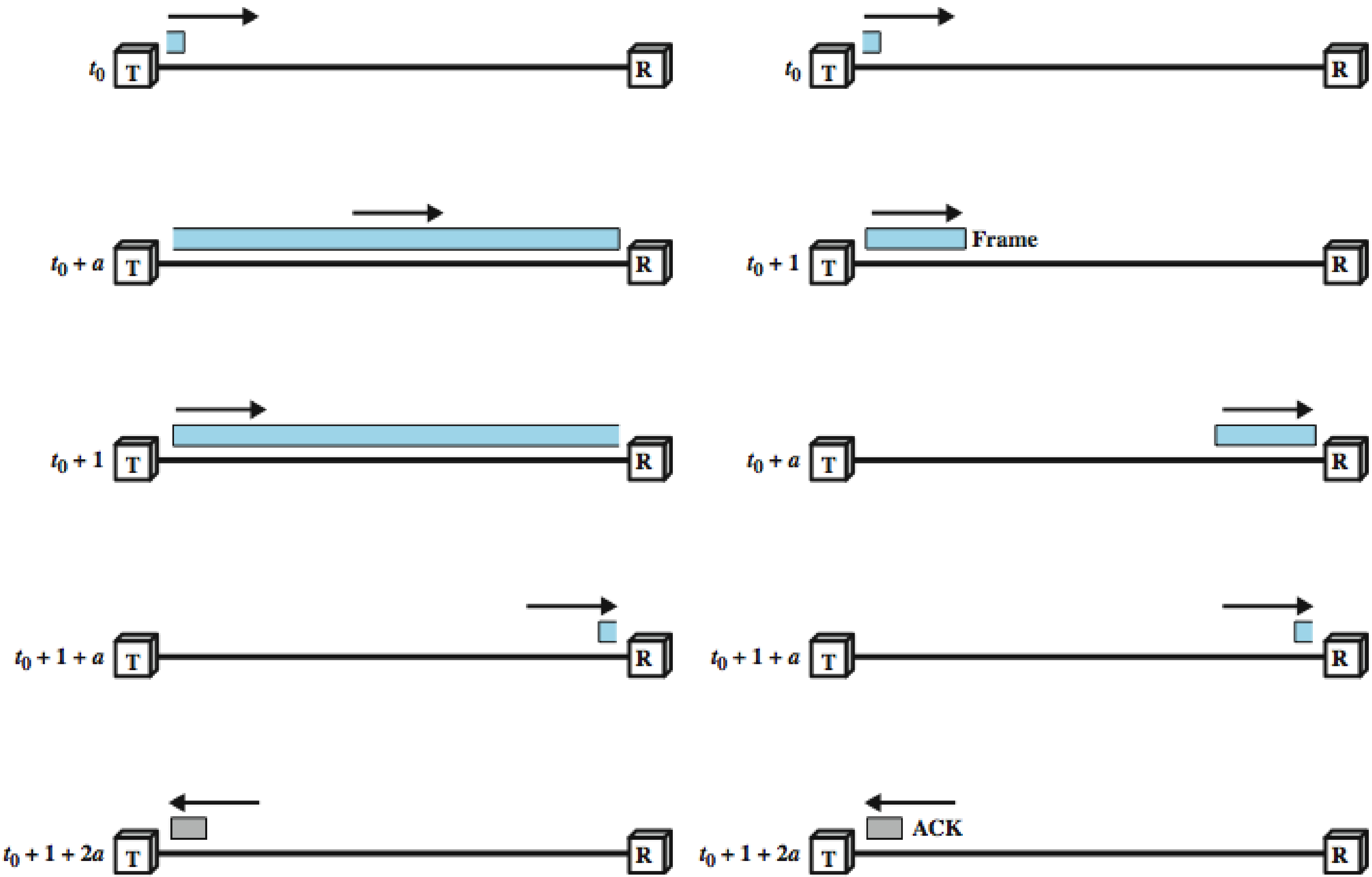
B = R x (d/V)

- **B: length of the link in bits**
- **R: data rate of the link, in bps**
- **d: length or distance of the link in meters**
- **V: velocity of propagation, in m/s**

a = B/L

- **a: the propagation delay (with frame tx time = 1)**
- **L: the number of bits in the frame**
- **Propagation delay=d/V**
- **Round-Trip Time = 2 × propagation delay**

Stop and Wait Link Utilization

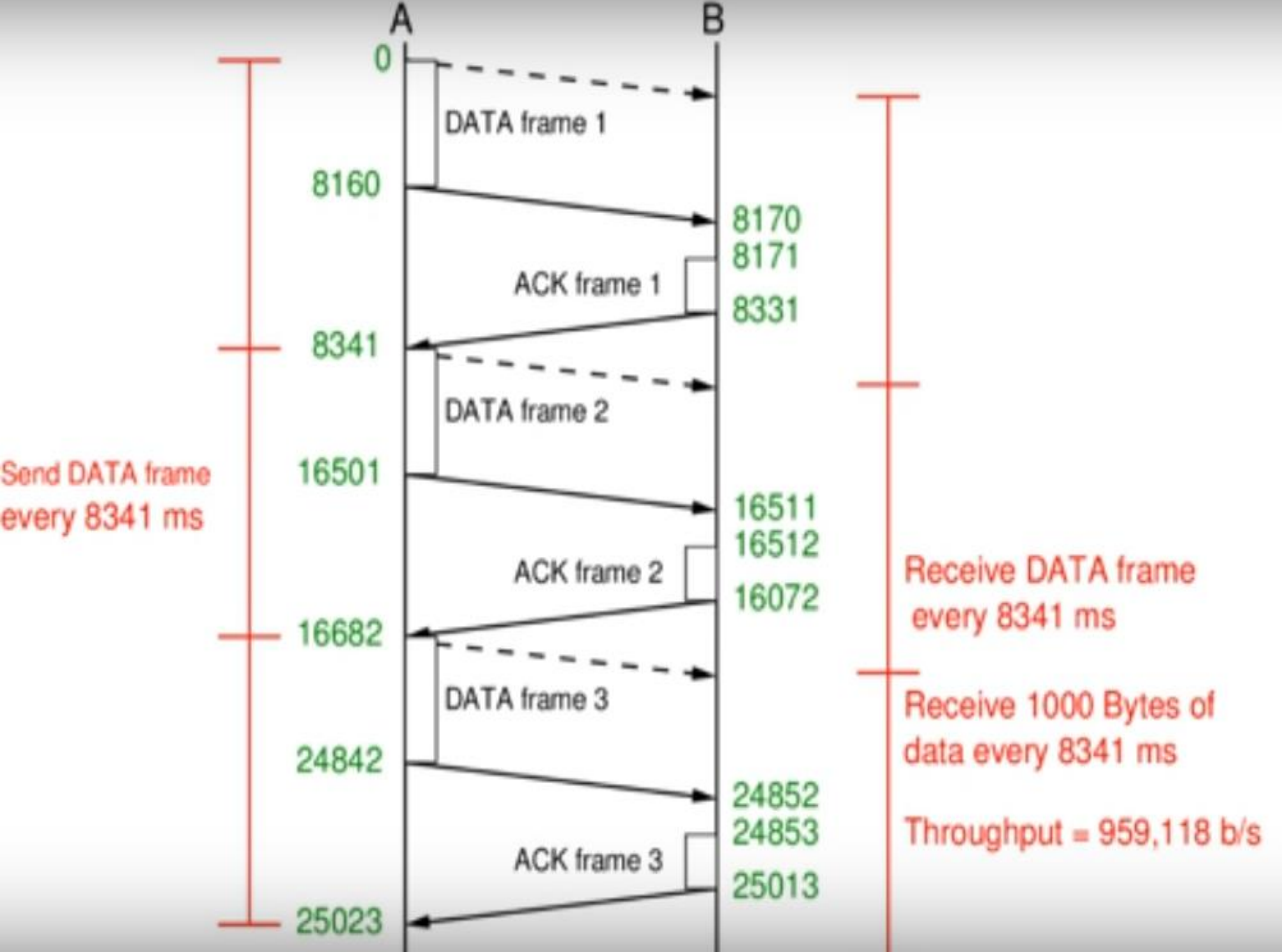


(a) $a < 1$

(b) $a > 1$

Example

Source has 3×1000 Byte messages to be sent immediately. Destination takes $1\mu s$ to process each frame. DATA frame contains 1000 Bytes data plus 20 Byte header. ACK frame is 20 Bytes. Link is 2km, 1Mb/s and velocity of $2 \times 10^8 m/s$. What is the throughput?



A
B
0
DATA frame 1
8160
8170
8171
ACK frame 1
8331
8341
DATA frame 2
16501
16511
16512
ACK frame 2
16072
16682
DATA frame 3
24842
24852
24853
ACK frame 3
25013
25023
Send DATA frame
every 8341 ms
Receive DATA frame
every 8341 ms
Receive 1000 Bytes of
data every 8341 ms
Throughput = 959,118 b/s

EXAMPLE 7.1 Consider a 200-m optical fiber link operating at 1 Gbps. The velocity of propagation of optical fiber is typically about 2×10^8 m/s. Using Equation (7.1), $B = (10^9 \times 200)/(2 \times 10^8) = 1000$ bits. Assume a frame of 1000 octets, or 8000 bits, is transmitted. Using Equation (7.2), $a = (1000/8000) = 0.125$. Using Figure 7.2a as a guide, assume transmission starts at time $t = 0$. After 1 μs (a normalized time of 0.125 frame times), the leading edge (first bit) of the frame has reached R, and the first 1000 bits of the frame are spread out across the link. At time $t = 8\ \mu$s, the trailing edge (final bit) of the frame has just been emitted by T, and the final 1000 bits of the frame are spread out across the link. At $t = 9\ \mu$s, the final bit of the frame arrives at R. R now sends back an ACK frame. If we assume the frame transmission time is negligible (very small ACK frame) and that the ACK is sent immediately, the ACK arrives at T at $t = 10\ \mu$s. At this point, T can begin transmitting a new frame. The actual transmission time for the frame was 8 μs, but the total time to transmit the first frame and receive an ACK is 10 μs.

Now consider a 1-Mbps link between two ground stations that communicate via a satellite relay. A geosynchronous satellite has an altitude of roughly 36,000 km. Then $B = (10^6 \times 2 \times 36{,}000{,}000)/(3 \times 10^8) = 240{,}000$ bits. For a frame length of 8000 bits, $a = (240{,}000/8000) = 30$. Using Figure 7.2b as a guide, we can work through the same steps as before. In this case, it takes 240 ms for the leading edge of the frame to arrive and an additional 8 ms for the entire frame to arrive. The ACK arrives back at T at $t = 488$ ms. The actual transmission time for the first frame was 8 ms, but the total time to transmit the first frame and receive an ACK is 488 ms.

What Size Frames To Use?

Protocols often limit size of packets (frames), i.e. maximum number of bytes of data or payload

Large frames minimize header overheads

Small frames:

- Allow more data to be sent when receive buffers are limited
- Introduce small overhead if a retransmission is required
- Allow fair sharing amongst multiple users

Optimal packet size depends on overheads, and desired throughput and delay performance

Efficiency of Stop-and-Wait Flow Control

- Efficiency n, for stop-and-wait flow control:

- where: Data, Hdr, Ack are transmission times of original data in DATA frame, header in DATA frame and ACK frame respectively; Prop is link propagation time

$$\eta = \frac{\mathsf{Data} + \mathsf{Hdr}}{\mathsf{Data} + \mathsf{Hdr} + \mathsf{Ack} + 2 \times \mathsf{Prop}}$$

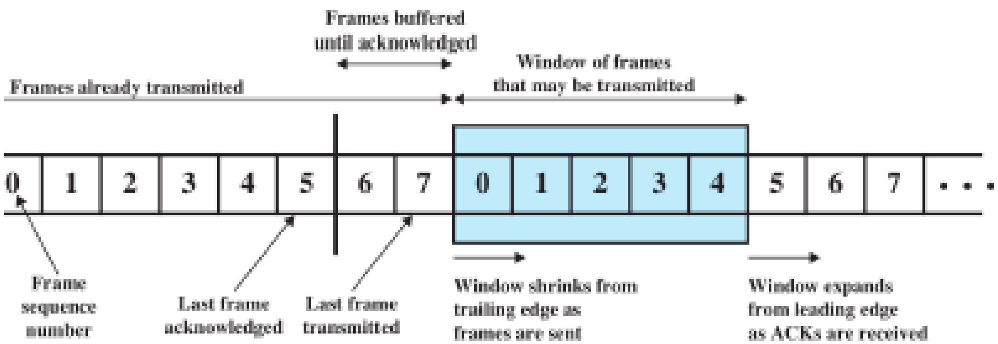
Sliding-Window Flow Control

Stop-and-wait allows only 1 frame to be in transit at a time

Sliding-window flow control allows multiple frames to be in transit at a time

Sequence Numbers

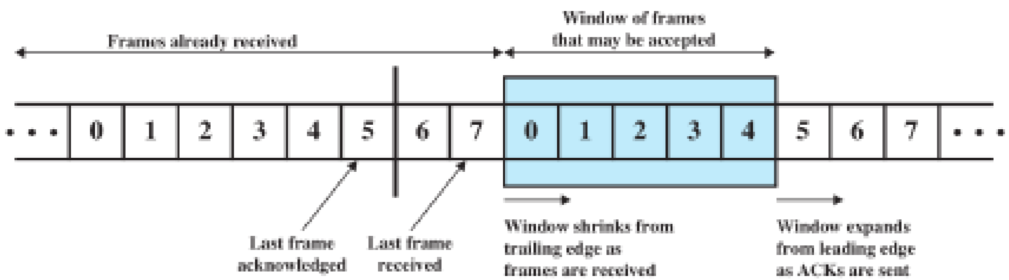
- Each frame header contains k-bit sequence number (wraps back to 0 after $2^k -1$)
- Keep track of frames sent and acknowledged



Sliding-Window Flow Control

The Sender

- Sender is allowed to send up to W frames without receiving ACK
- Sender records:
  - Last frame acknowledged
  - Last frame transmitted
  - Current window size

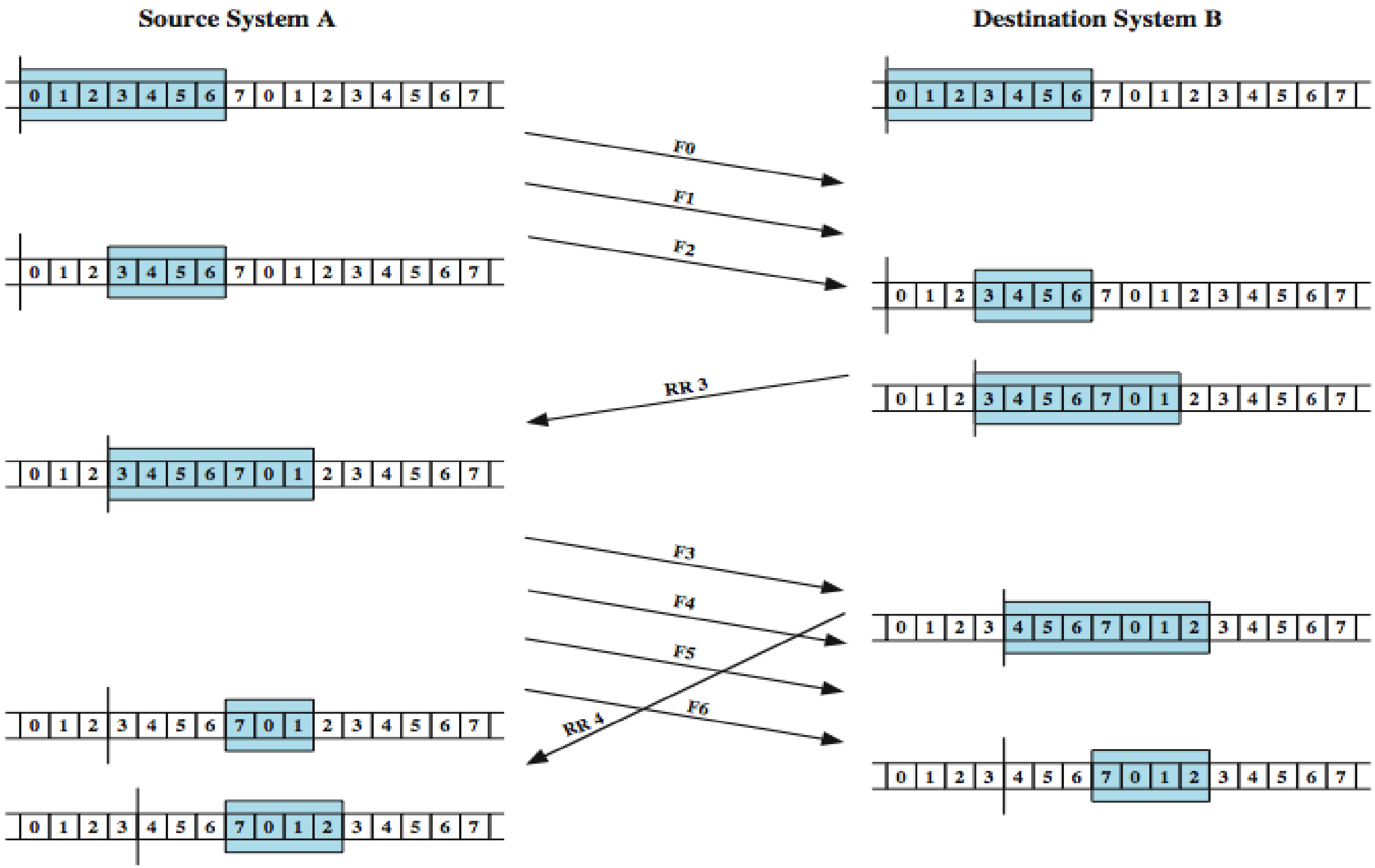


The Receiver

- Receiver has buffer space for W frames
- Receiver records:
  - Last frame acknowledged
  - Last frame received
  - Current window size
- Receiver sends an ACK (or Receiver Ready, RR) frame
- ACK contains sequence number of next expected DATAframe

Sliding-Window Flow Control

Example



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

EXAMPLE 7.3 Let us consider the use of sliding-window flow control for the two configurations of Example 7.1. As was calculated in Example 7.1, it takes 10 μs for an ACK to the first frame to be received. It takes 8 μs to transmit one frame, so the sender can transmit one frame and part of a second frame by the time the ACK to the first frame is received. Thus, a window size of 2 is adequate to enable the sender to transmit frames continuously, or at a rate of one frame every 8 μs. With stop-and-wait, a rate of only one frame per 10 μs is possible.

For the satellite configuration, it takes 488 ms for an ACK to the first frame to be received. It takes 8 ms to transmit one frame, so the sender can transmit 61 frames by the time the ACK to the first frame is received. With a window field of 6 bits or more, the sender can transmit continuously, or at a rate of one frame every 8 ms. If the window size is 7, using a 3-bit window field, then the sender can only send 7 frames and then must wait for an ACK before sending more. In this case, the sender can transmit at a rate of 7 frames per 488 ms, or about one frame every 70 ms. With stop-and-wait, a rate of only one frame per 488 ms is possible.

Example

Source always has data ready to send; DATA frame contains 9000 bits payload, 1000 bits header; ACK frame is 1000 bits; data rate 100kb/s; propagation 200ms. What is efficiency if using 2-bit sequence number? 3-bits?

Utilization

- $U=1$ when $W \geq 2a+1$
- $U=W/2a+1$ when $W<2a+1$

Additional Features of Sliding Window

- Receive Not Ready frame: acknowledges received frames but does not allow any more data
- Piggybacking: DATA frame header contains sequence number of DATA and sequence number of ACK (acknowledgement number)
- If no DATA to send, normal ACK is transmitted
- If no new ACK, previous ACK number is repeated in DATA frame

Example

- A channel has a data rate of 4 kbps and a propagation delay of 20 ms. For what range of frame sizes does stop-and-wait give an efficiency of at least 50%?

Stop-and-Wait Flow Control

1. Station sends F1
2. Station sends an acknowledgment.
3. Station sends F2
4. Station sends an acknowledgment.
5. Station sends Fn
6. Station sends an acknowledgment.

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

Let us assume that the processing time is relatively negligible, and that the acknowledgment frame is very small compared to a data frame, both of which are reasonable assumptions. Then we can express the total time to send the data as

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

The utilization, or efficiency, of the line is

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

It is useful to define the parameter $a = t_{prop}/t_{frame}$ (see Figure 7.2). Then

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

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

$$a = \frac{d/V}{L/R} = \frac{Rd}{VL}$$

$$a = \frac{\text{Propagation Delay}}{\text{Transmission Time}} = \frac{20 \times 10^{-3}}{L/\left(4 \times 10^{3}\right)} = \frac{80}{L}$$

$$U = \frac{1}{1+2a} = \frac{1}{1+\left(160/L\right)} \geq 0.5$$

$$L \geq 160$$

Therefore, an efficiency of at least 50% requires a frame size of at least 160 bits.

Error Control

Need to detect and correct errors such as:

- Lost frames: frame not received
- Damaged frames: frame received with errors

Common techniques used:

- Error detection and FEC
- **Positive acknowledgment**: destination returns a positive ACK after successfully receiving error-free frames
- **Retransmission after timeout**: source retransmits a frame that has not been ACKed after predetermined time
- **Negative acknowledgement and retransmission**: destination returns negative ACK for frames in which an error is detected

- Last 3 techniques are called automatic repeat request (ARQ). Three versions:
  - Stop-and-wait ARQ
  - Go-back-N ARQ
  - Selective-reject ARQ

Stop-and-Wait ARQ

Based on Stop-and-Wait flow control

Source transmits single frame, starts timer and maintains copy

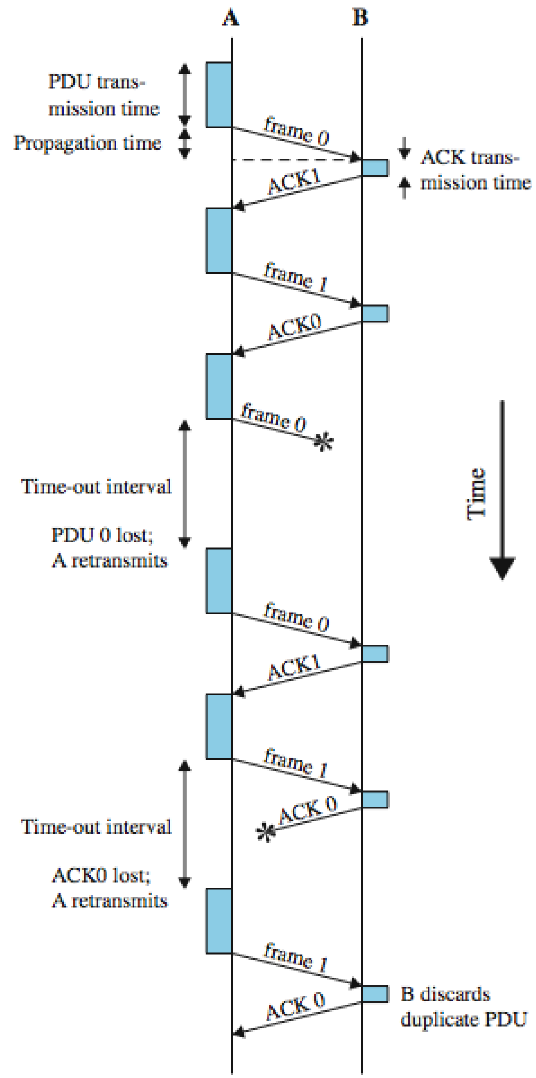
- If ACK received, stop timer and transmit next frame
- If no ACK received before timer expires, retransmit copy of frame

Destination sends ACK if frame received (with no errors); if damaged frame, then discard frame

Frames have 1-bit sequence number (alternate between 0 and 1)

- Used for destination to distinguish between duplicate DATA frames in case of damaged ACK

Stop-and-Wait ARQ is simple, but inefficient


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

Go-Back-N ARQ

- Based on Sliding Window flow control
- If no error, ACK as in sliding window (contains sequence number of next expected frame)
- If error detected by Destination, reply with negative ACK (NACK or rejection, REJ)
- Destination will discard that frame and all future frames until error frame received correctly
- Transmitter must go back and retransmit that frame and all subsequent frames
- If no response from Destination after timeout, then Source may send special ACK (ACKRequest or RR(P bit = 1))
  - The ACKRequest from Source to Destination, is a request for an ACK from the Destination
  - Upon receipt of ACKRequest, the Destination sends an ACK
- Maximum window size: **2^k -1**

Go Back N - Handling

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.*

Go Back N - Handling

1. Damaged Frame

- error in frame *i* so receiver rejects frame *i*
- transmitter retransmits frames from *i*

(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 - Handling

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 (i+1) 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

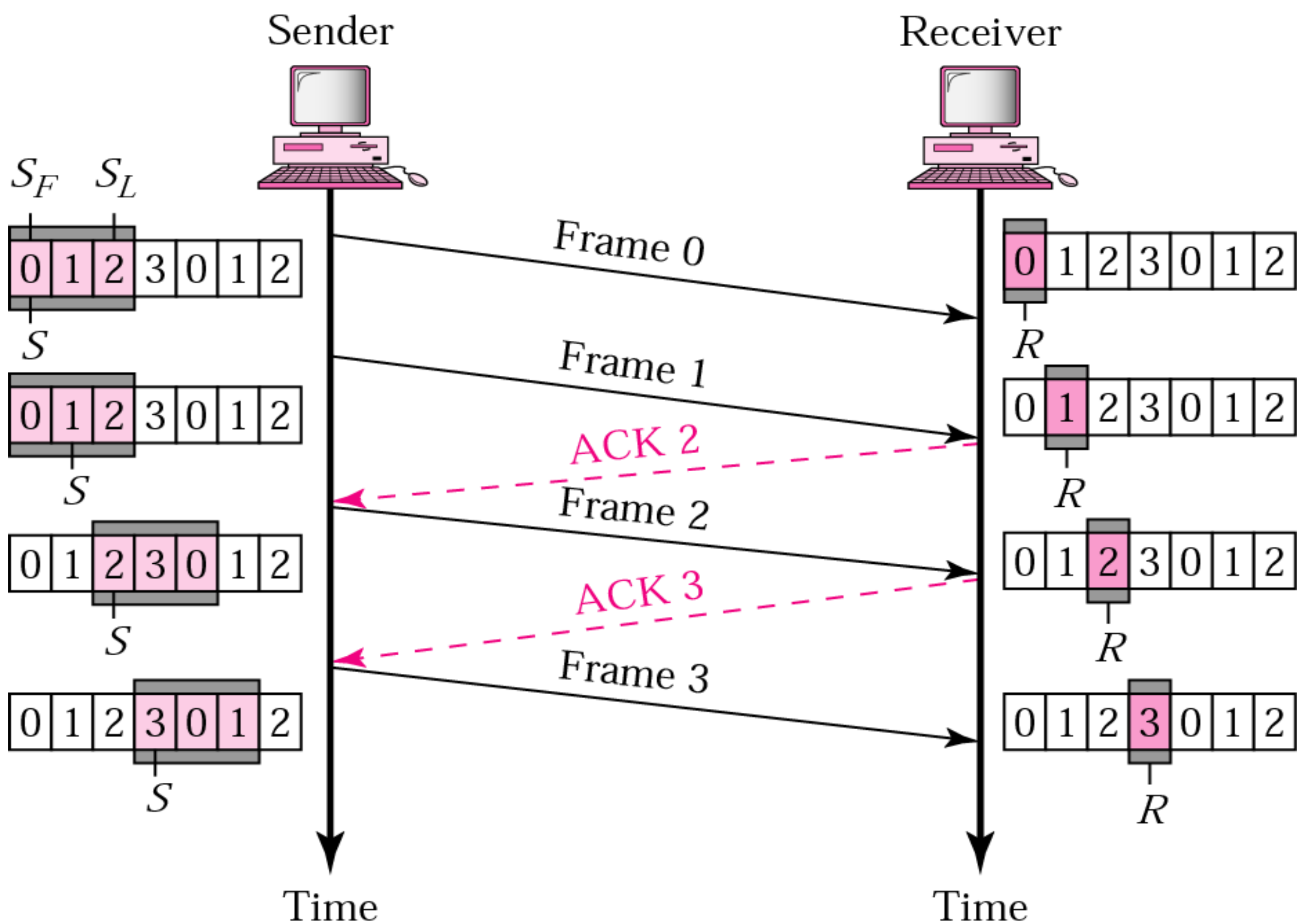
Go Back N

- **Setup – Sender sliding window**
- Frames are **numbered sequentially** using **m-bit** in the the **frame heade**r
  - For m bits frame sequence numbers 0 – (2^m –1), repeated
  - m=3 ➔ (0,1,2,3,4,5,6,7,0, 1,2,3,4,5,6,7,…)
- Sender can send multiple frames while waiting for ACK, but **total number of unacknowledged frames** should **not** exceed **2^m –1** which is called **window**
- **Window** is an **imaginary placeholder** that covers the frames sequence numbers which can be in **transit**
- The frames to **the left** of the window are already **acknowledged** and to **the right** can not be sent until window slides over them
- Frames **inside** the window are **outstanding** (have been sent but not yet acknowledged OR **will be sent soon once** received from upper-layer)
- **Sender window will slide to frame number (i) when (ACK i) is received.**

Go Back N - Handling

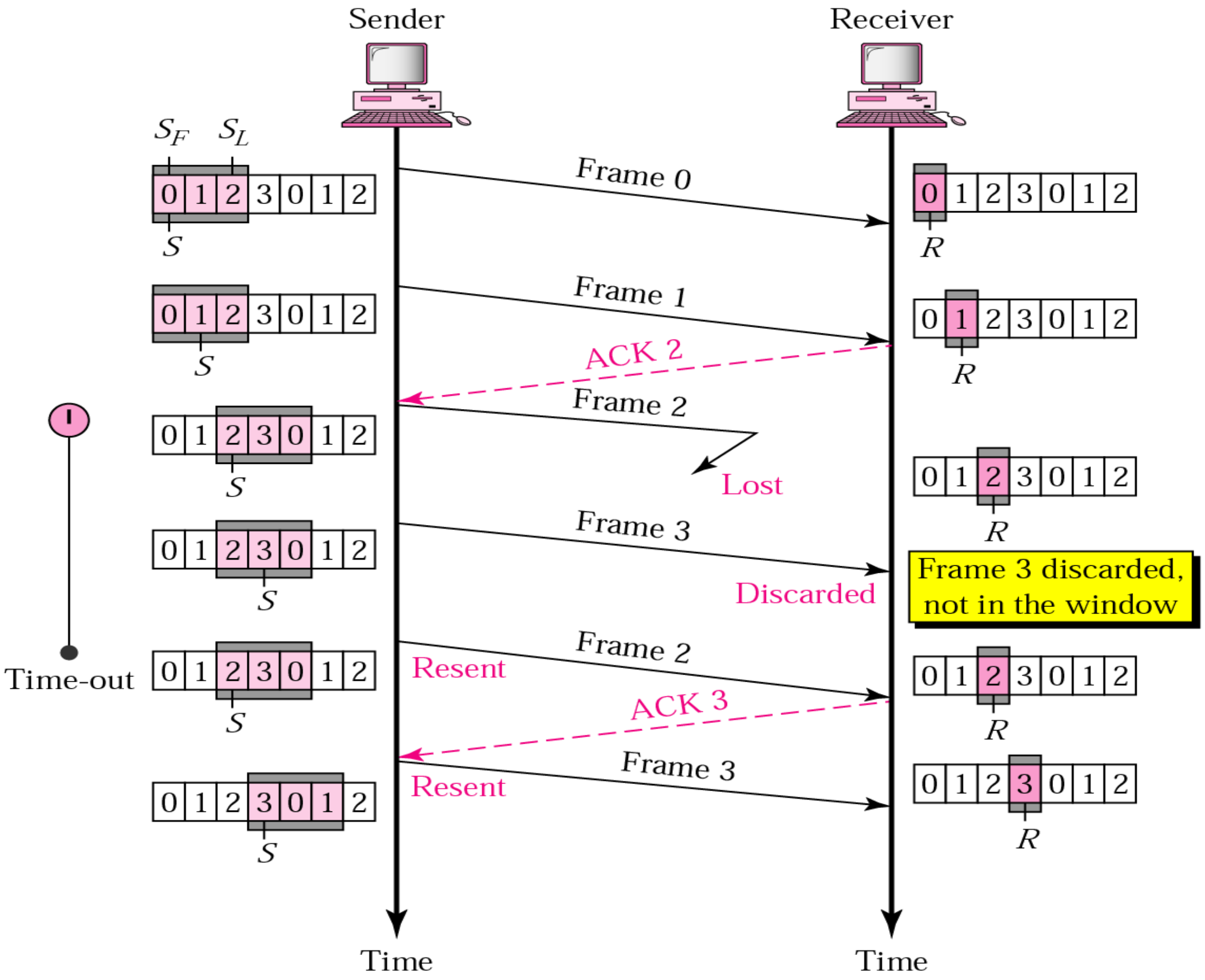
- **Setup –Receiver sliding window**
- The size of the receiver window is always **1** and points to the **next expected frame** number to arrive
- This means that frames should arrive **in order**
- If the expected frame is received without errors, the receiver window slides over the **next sequence number**.
- **Operation**
- The receiver sends a positive ACK if a frame has arrived **without error** and **in order** (with the expected sequence number)
- Receiver does not have to acknowledge each individual frame received correctly and in order.
- Receiver can send **cumulative ACK** for several frames (**ACK 5 acknowledges frames (0,1,2,3,4) and expecting frame 5)**
- If the frame is **damaged or out-of-order**, the receiver **discards it** (and **stay *silent***) and also **discards all subsequent frames** until it receives the one expected.
  - In this case, **no** ACK will be transmitted
- If the sender timer expires before receiving an ACK, it will **resend *ALL* frames beginning with the one expired until the last one sent** (Go-Back-N).

Go-Back-N ARQ, normal operation

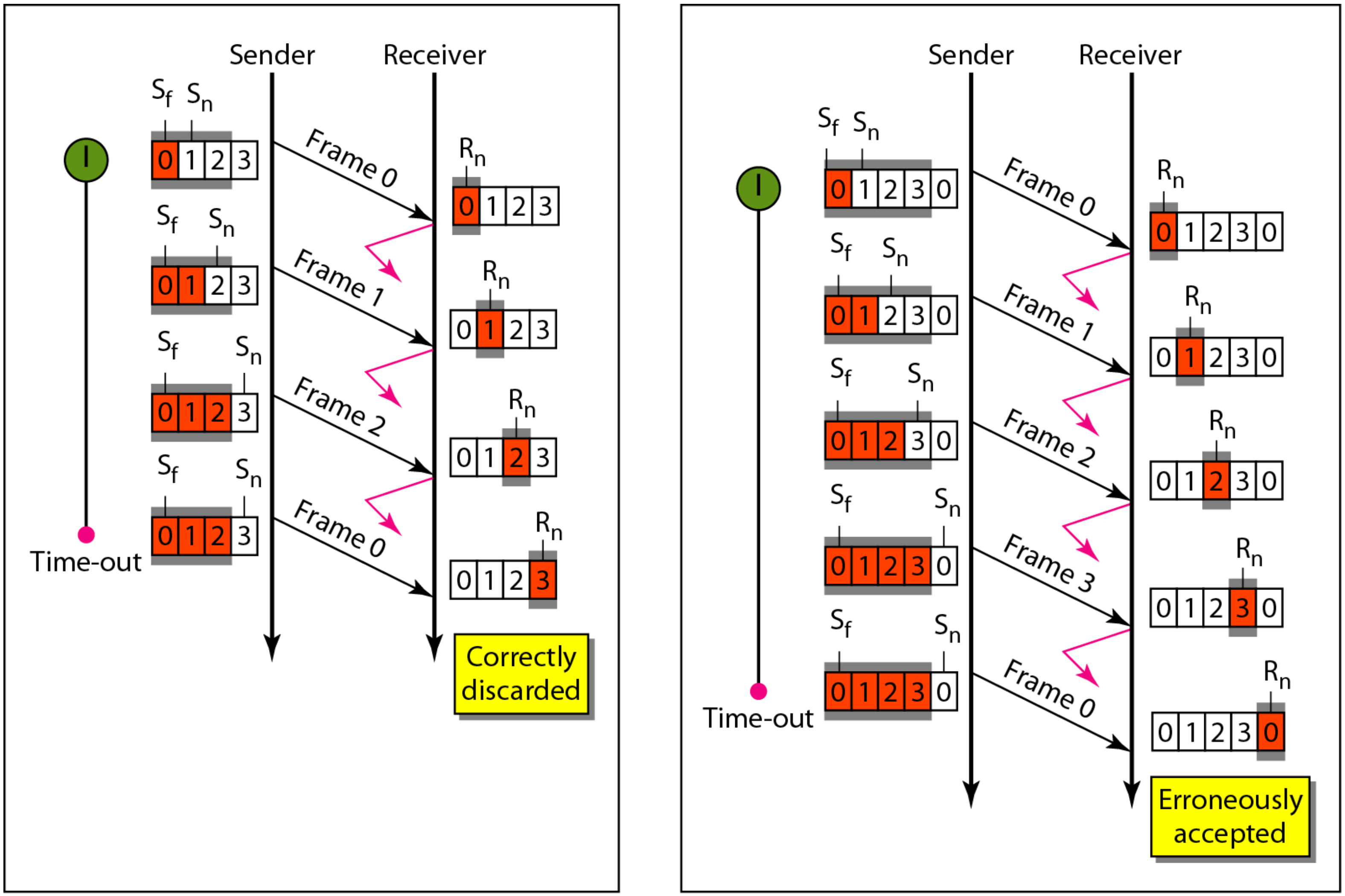


Sender window will slide to frame number (i)
when (ACK i) is received.

Go-Back-N ARQ, lost frame



Window size for Go-Back-N ARQ



a. Window size $< 2^m$

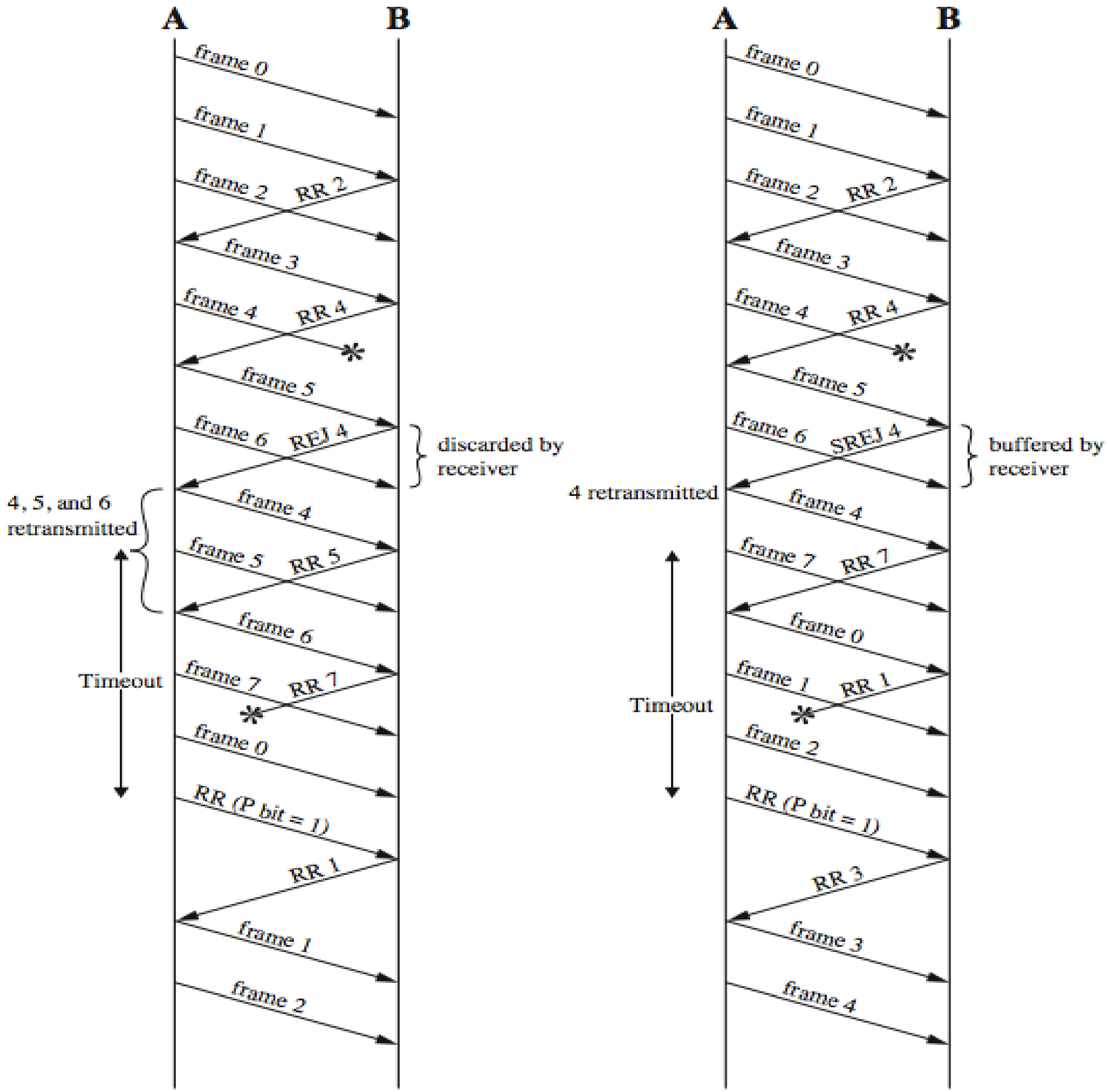
b. Window size $= 2^m$

Performance

Better than Stop and wait because it keeps the sender busy while waiting for acknowledgement

Does not allow error free but out of order frames to be accepted by the receiver

This protocol **can waste a lot of bandwidth** if the error rate is high because it requests the sender to retransmit the frame in error and all the subsequently transmitted frames



(a) Go-back-N ARQ

(b) Selective-reject ARQ

Selective-Reject ARQ

- Also called selective retransmission or selective repeat
- Only frames that are rejected or timeout are retransmitted
- Subsequent frames are accepted by the destination and buffered
- Maximum window size: 2^{k-1}
- Minimizes retransmission (GOOD)
- Destination must maintain large enough buffer for frames received out- of-order (BAD)
- More complex logic in transmitter (BAD)
- Not as widely used as Go-Back-N; useful for satellite links with long propagation delays

Selective-Reject ARQ

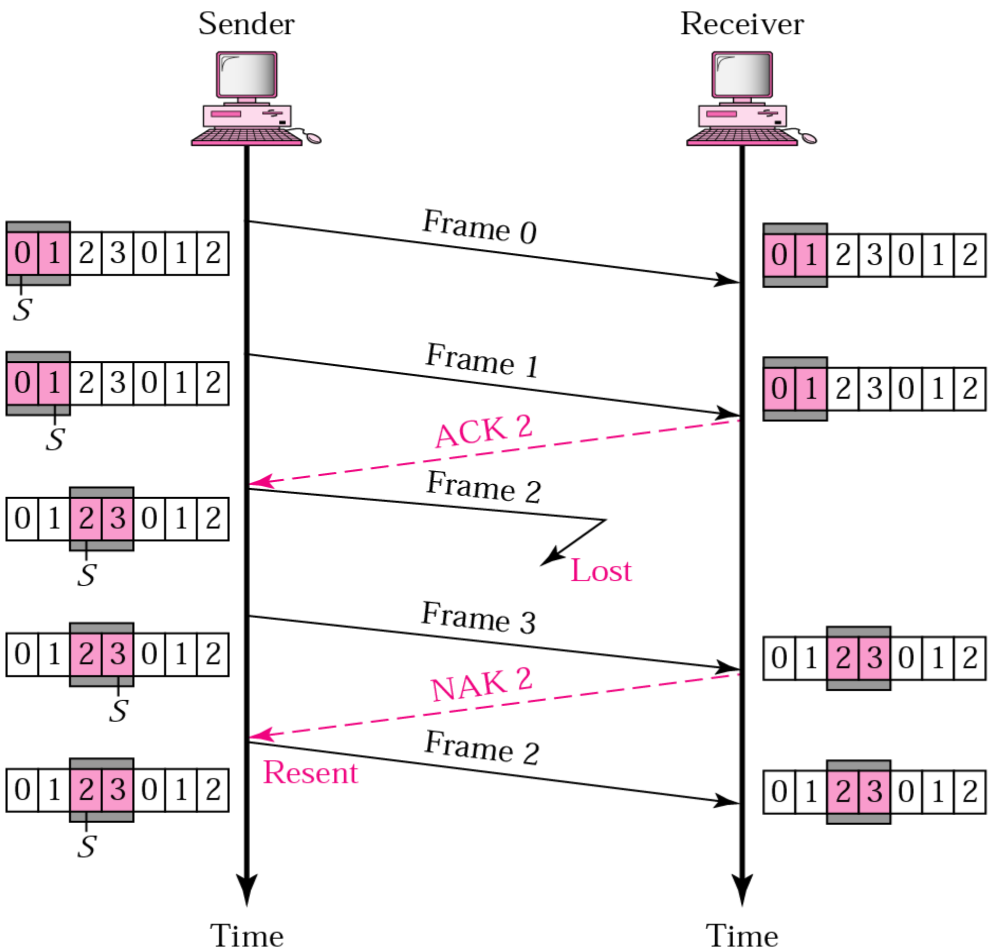
Setup-Window size (for both sender and receiver)

- For m bit sequence number **the maximum** window size $2^{m\text{-}1}$

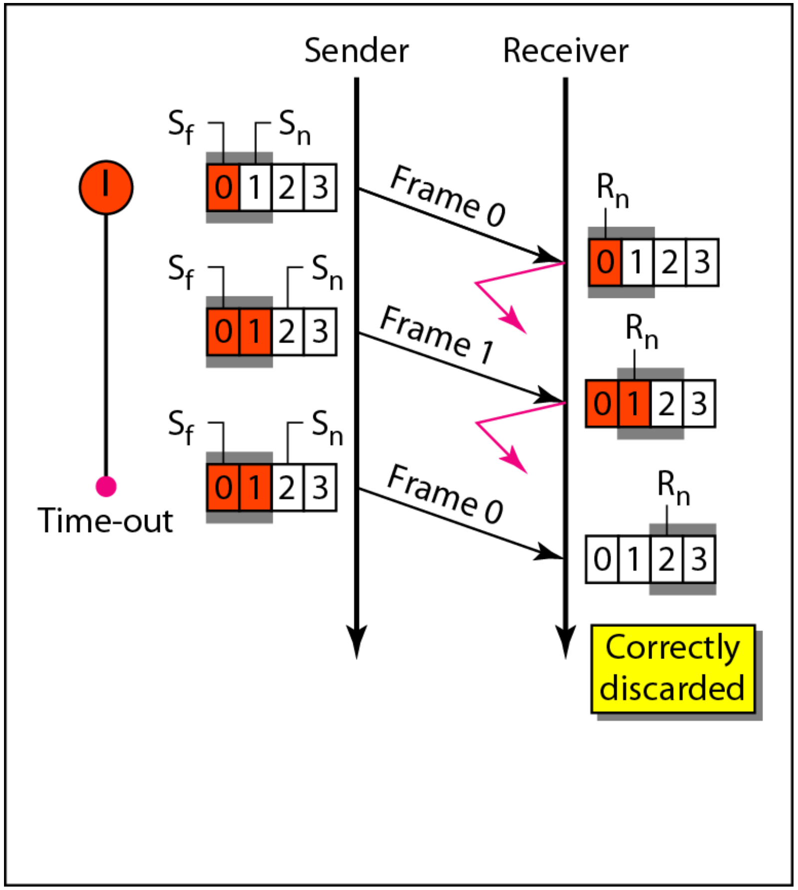
Operation

- Sender in case of **damaged or lost frame** retransmits
  - Those which are <u>**negatively acknowledged**</u>
  - Those for <u>which timer expires</u>
- Receiver can **accept out of order frames** and buffer (store) them **until the lost, damaged, or delayed frame arrives.**
- Receiver **does not acknowledge out of order frames** but **buffers** them only
- If the receiver receives an **out-of-order, error free frame**, it will send a frame called **Negative Acknowledge (NAK)** with the number of the frame to be retransmitted only.
- NAK improves the performance because it requests retransmission of the lost frame **before** the corresponding sender timer expires

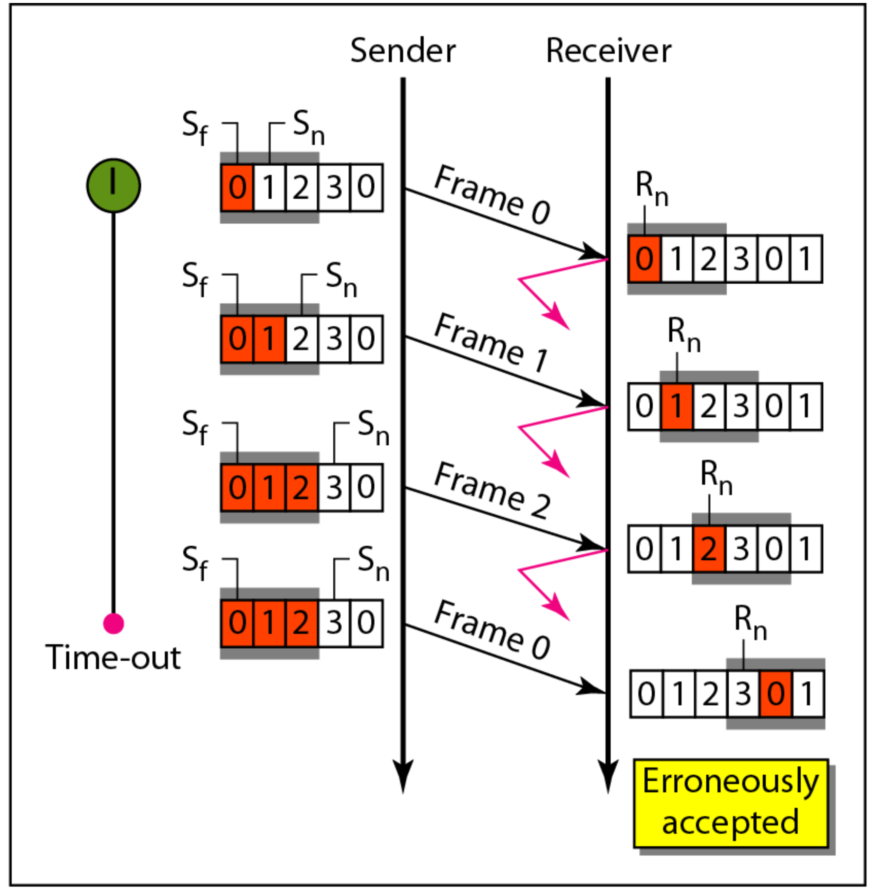
Selective Repeat ARQ, lost frame



Selective Repeat ARQ, window size



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



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

Performance

More efficient than the other two protocols because it **reduces number of retransmissions** for noisy links

The receiver and transmitter processing logic is more complex

- **Receiver** must be able to **reinsert** the retransmitted (lost, delayed, damaged) frame in the proper sequence after it arrives
- **The sender** should be able to **send out of order** frame when requested by the sender using NAK
- Needs **more memory** than Go-Back-N ARQ at the **receiver side**. The receiver memory is large if the window size is large

Example

- 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?

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

$L=10^3$, $R=10^6$

- 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

- 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 2×10^8 m? Ignore transmission, waiting, and processing delays. We 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

- Repeat using the *Go-back-N* ARQ Protocol with a window size of 7.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

Example

- 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

- 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**.