Protocols with pipelining

Sachin Gajjar

sachin.gajjar@nirmauni.ac.in

Pipelining

- In networking and in other areas, a task is often begun before the previous task has ended.
- This is known as pipelining.
- several frames can be sent before we receive ACK about the previous frames.
- Improves efficiency of transmission if number of bits in transition is large with respect to BDP
- Two protocols
  - Go Back N ARQ
  - Selective Repeat ARQ

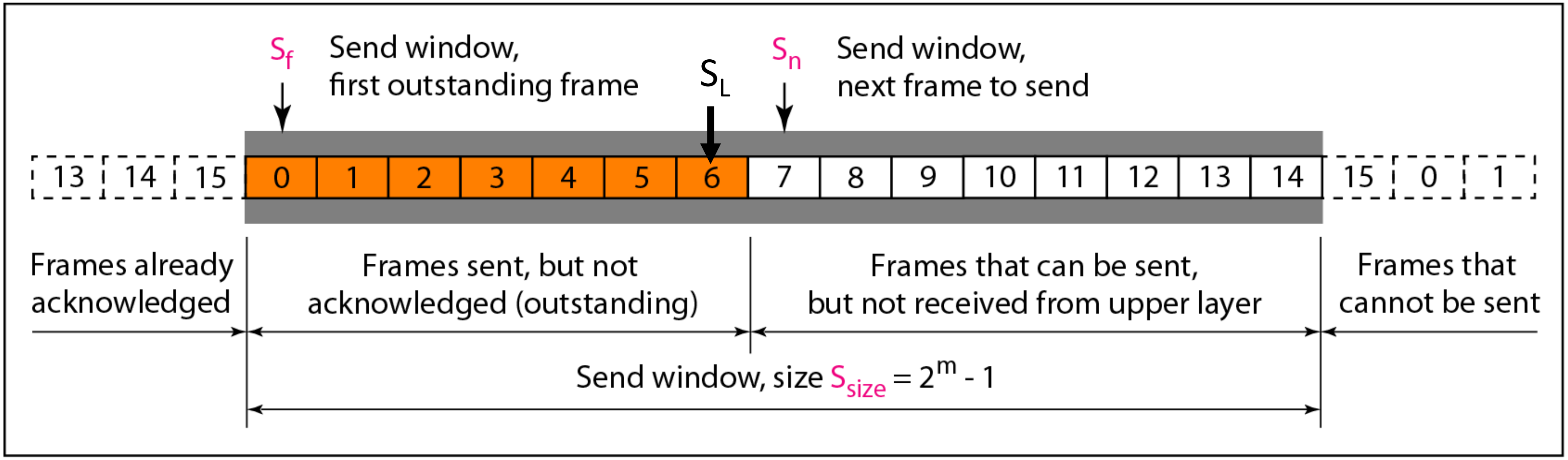
Go-Back-N Automatic Repeat Request

- Keep channel busy by sending more than 1 frame
- send several frames before receiving ACK
- Keep copy of frames until ACK comes
- Frames from sender are numbered sequentially
- Sequence number limit = number of bits for sequence number in header
- m bits → sequence no. range = 0 to $2^m - 1$
- Seq. Nos. = 0 to $2^m - 1$ and repeat

Go-Back-N ARQ Sliding Window

- an abstract concept that defines range of seq. nos. that is the concern of sender and receiver
- sx/rx need to deal with only part of seq. nos.
- sender range = send sliding window
- receiver range = receive sliding window
- Send window is an imaginary box covering sequence numbers of data frames sx can send

Go-Back-N ARQ
Sliding window of size 15 (m = 4)

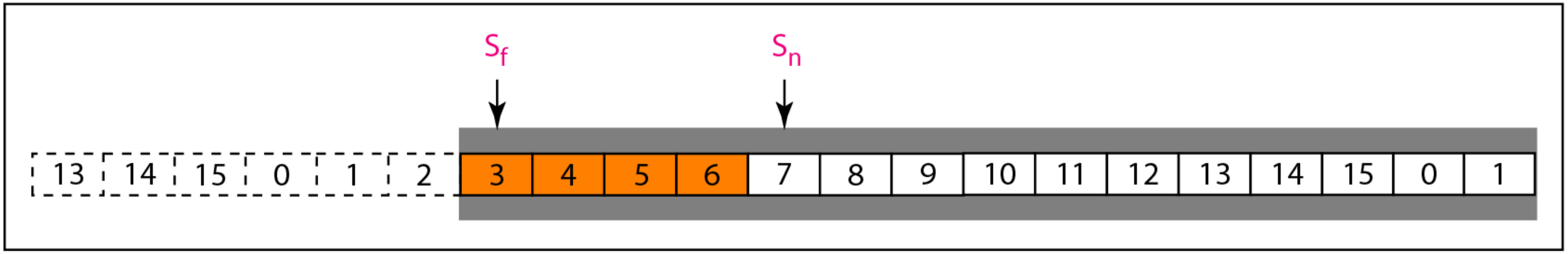

S_f
Send window,
first outstanding frame
S_L
S_n
Send window,
next frame to send
13
14
15
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
0
1
Frames already acknowledged
Frames sent, but not acknowledged (outstanding)
Frames that can be sent, but not received from upper layer
Frames that cannot be sent
Send window, size $S_{size} = 2^m - 1$

a. Send window before sliding

Send window is an abstract concept defining an imaginary box of size $2^m - 1$ with three variables: S_f, S_n, S_L and S_{size}

S_L = last outstanding frame

Go-Back-N ARQ Sliding window of size 15 (m = 4)



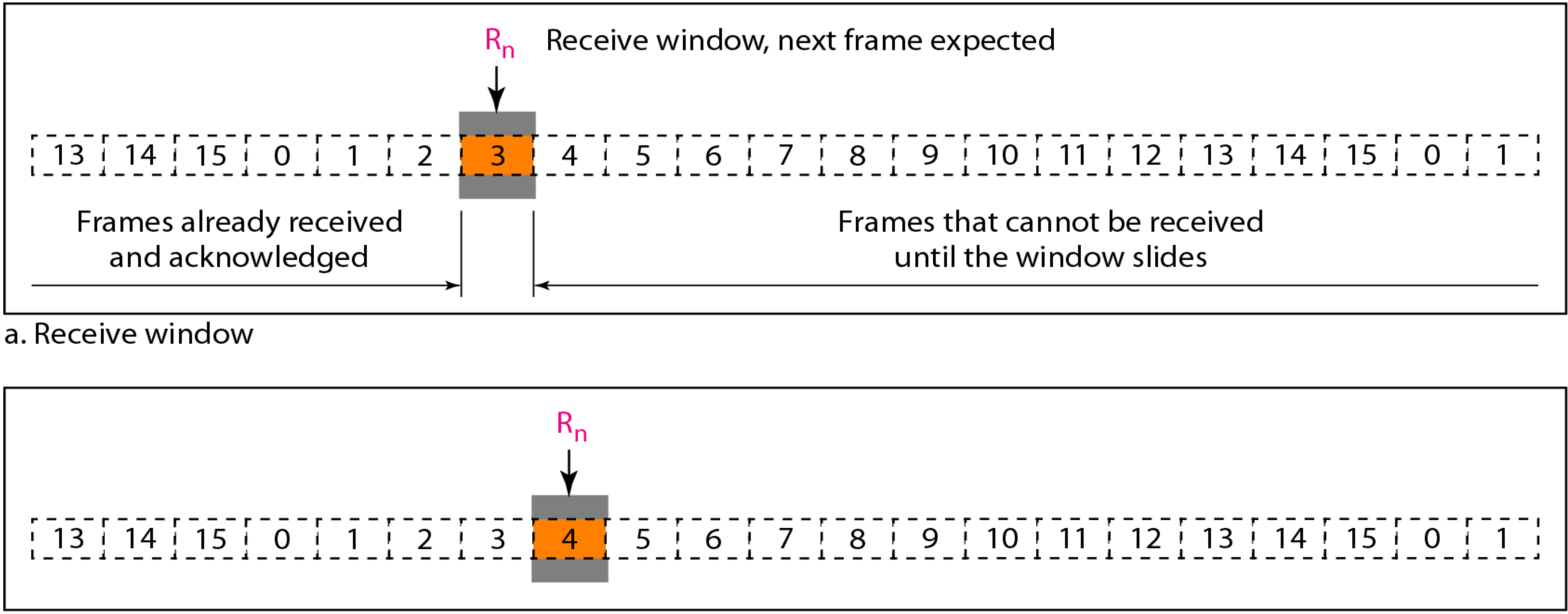
b. Send window after sliding

- send window can slide one or more slots to the right when ACK arrives from rx
- ACKs are cumulative more than one frame is acked
- frames 0, 1, and 2 are ACKed
- window slides to right 3 slots
- value of S_f = 3 as frame 3 is now first outstanding frame
- Send window can slide one or more slots when a valid ACK arrives

Go-Back-N ARQ Rx side

- rx window makes sure that correct frames are received and correct ACK are sent
- size of rx window = 1
- rx is always looking for arrival of a specific frame
- frame arriving out of order is discarded and needs to be resent

Go-Back-N ARQ Rx side


R_n
Receive window, next frame expected
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1
Frames already received and acknowledged
Frames that cannot be received until the window slides
R_n
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1

a. Receive window

b. Window after sliding

- rx window is an abstract concept defining an imaginary box of size 1 with variable R_n
- window slides when a correct frame has arrived
- sliding occurs one slot at a time

Go-Back-N ARQ Timers

- Can have timers for each frame sent
- However we can keep only one for all frames are sent
- As timer for first outstanding frame always expires first
- Send all outstanding frames when this timer expires

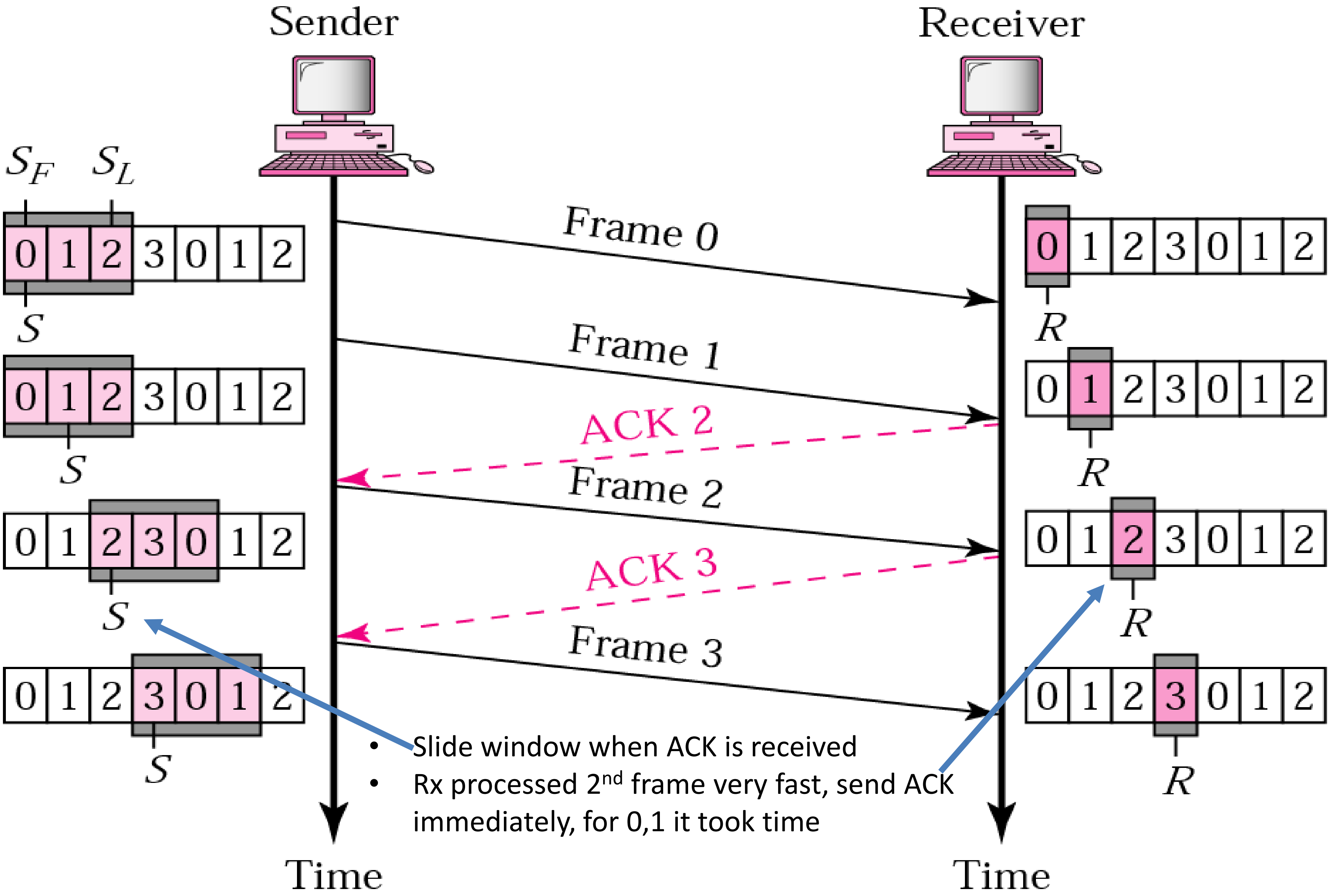
Go-Back-N ARQ Acknowledgment

- rx sends +ve ACK if frame has arrived safe/in order
- If frame is damaged/out of order rx is silent
- discards all subsequent frames until it gets reqd. one
- timer of unacked frame at sx expires
- Sx goes back and resends all frames, beginning with one with the expired timer
- rx does not have to ACK each frame received
- Sends cumulative ACK for several frames

Go-Back-N ARQ Resending a Frame

- When timer expires, sx resends all outstanding frames
- Suppose sx has already sent frame 6
- But timer for frame 3 expires (frame 3 has not been ACKed)
- sx goes back and sends frames 3, 4, 5, and 6 again
- i.e Go-Back-N ARQ

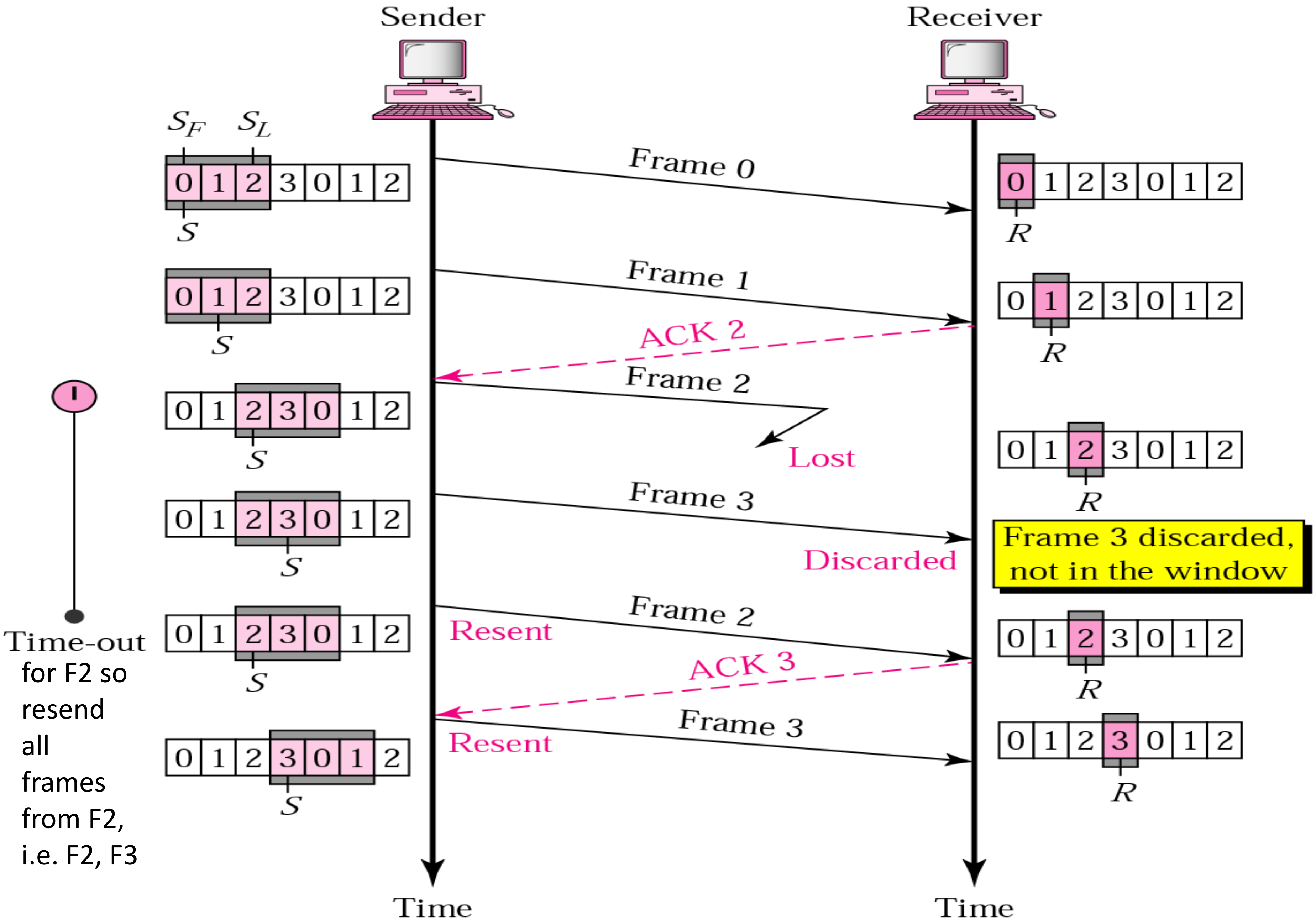
Go-Back-N ARQ, normal operation



- Slide window when ACK is received
- Rx processed 2nd frame very fast, send ACK immediately, for 0,1 it took time

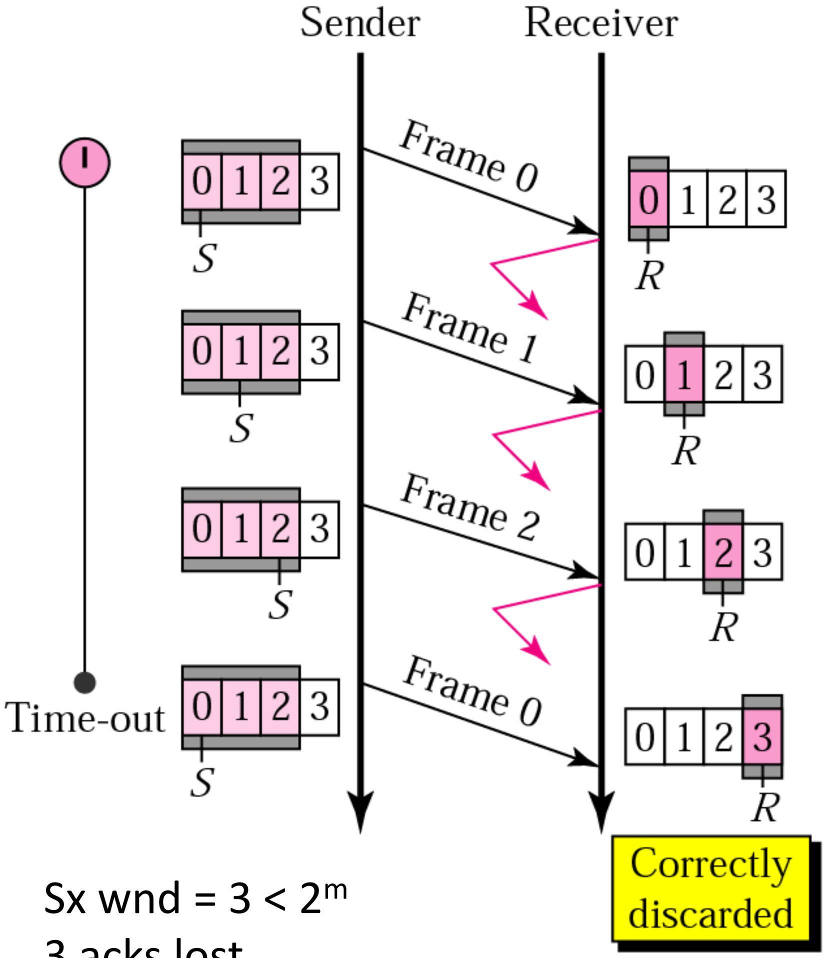
Go-Back-N ARQ, lost frame

Resent all frames in window from frame which is lost



for F2 so resend all frames from F2, i.e. F2, F3

m in header = 2, size of sx wnd can be $2^m - 1 = 3$



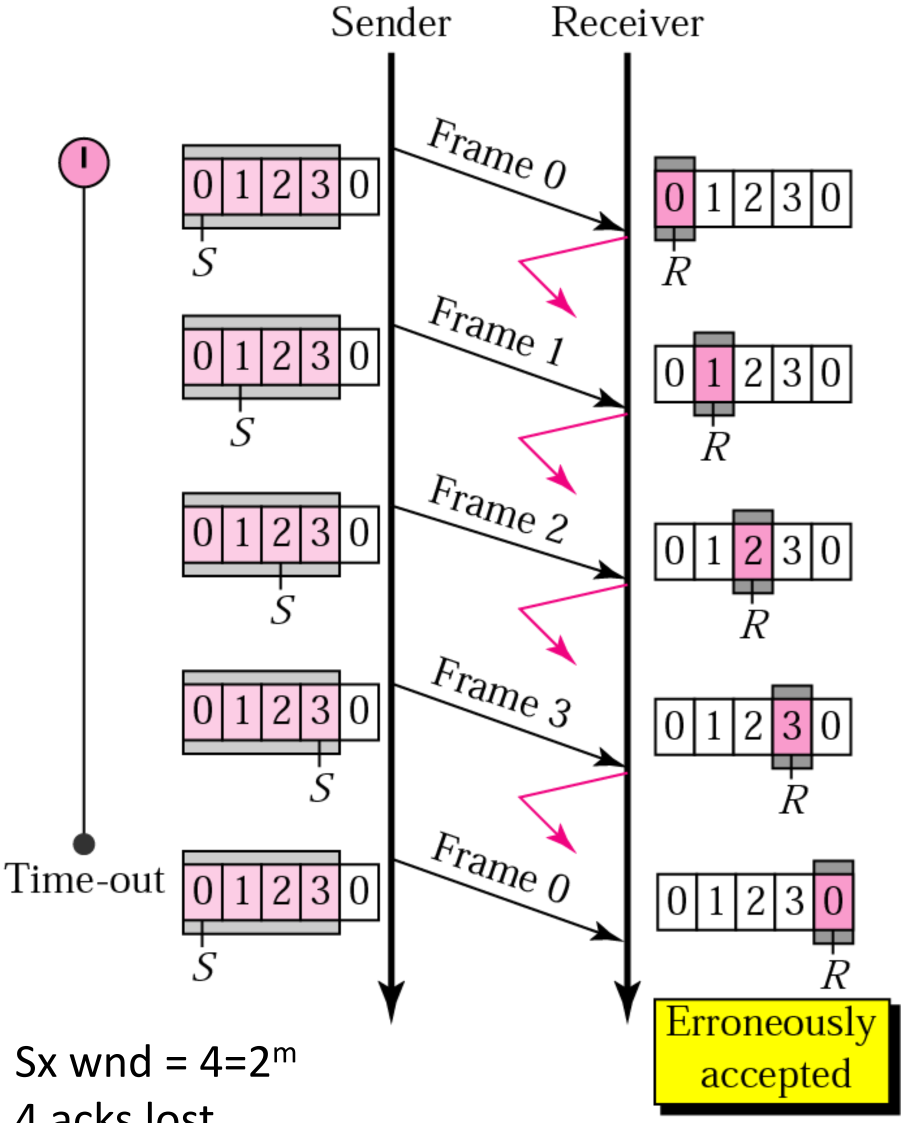
Sx wnd = 3 < 2^m

3 acks lost

F0 timer exp

F0 discared correctly

a. Window size < 2^m



Sx wnd = 4=2^m

4 acks lost

F0 timer exp

F0 accepted

b. Window size = 2^m

F0 Accepted again

Note

In Go-Back-N ARQ, the size of the send window must be less than 2^m;
the size of the receiver window
is always 1.

Note

Stop-and-Wait ARQ is a special case of Go-Back-N ARQ in which the size of the send window is 1.

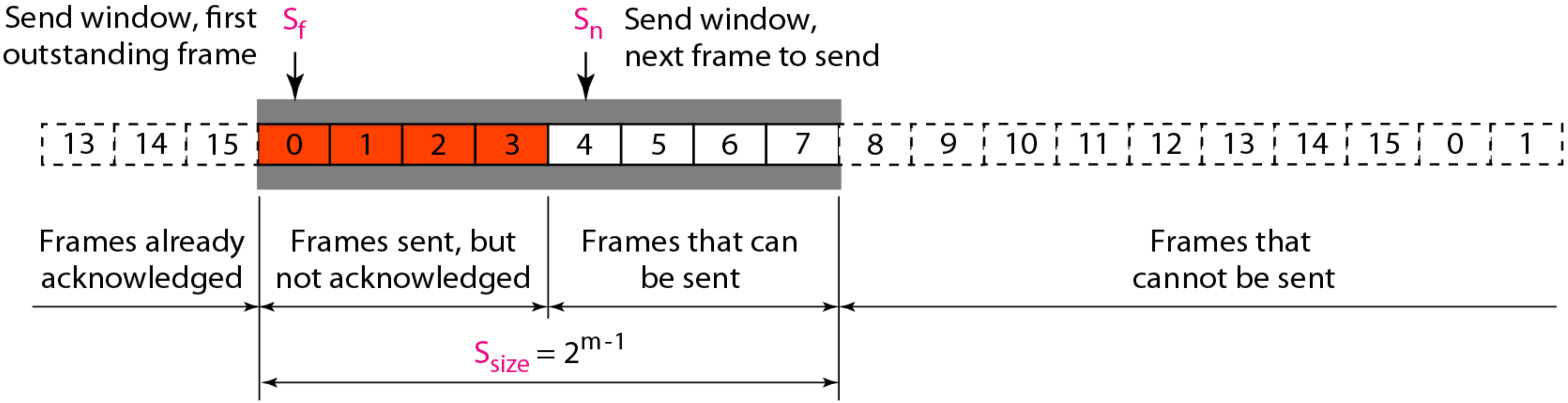
Go-Back-N ARQ

- simplifies the process at rx
- rx keeps track of one variable
- no need to buffer out-of-order frames, if received they are discarded
- very inefficient for noisy link
- More frames damaged →resending of multiple frames →uses up BW and slows down transmission
- For noisy links, only damaged frame is resent in Selective Repeat ARQ → processing at rx is complex

Selective Repeat ARQ Windows

- send wnd = receive wnd = 2^{m-1}
- m = 4, the seq. nos. of frames can be 0 to 15 (2^4-1)
- But size of wnd = 8 (2^{4-1}) (is 15 (2^4-1) in Go-Back-N)
- smaller wnd size → less efficiency in filling the pipe, but there are fewer duplicate frames
- same variables as for Go-Back-N
- Protocol allows as many frames as size of rx wnd to arrive out of order
- Rx arranges them and gives to network layer
- Rx sends ACK/NACK (lost frames)

Send window for Selective Repeat ARQ



S_f Send window, first outstanding frame

S_n Send window, next frame to send

Frames already acknowledged

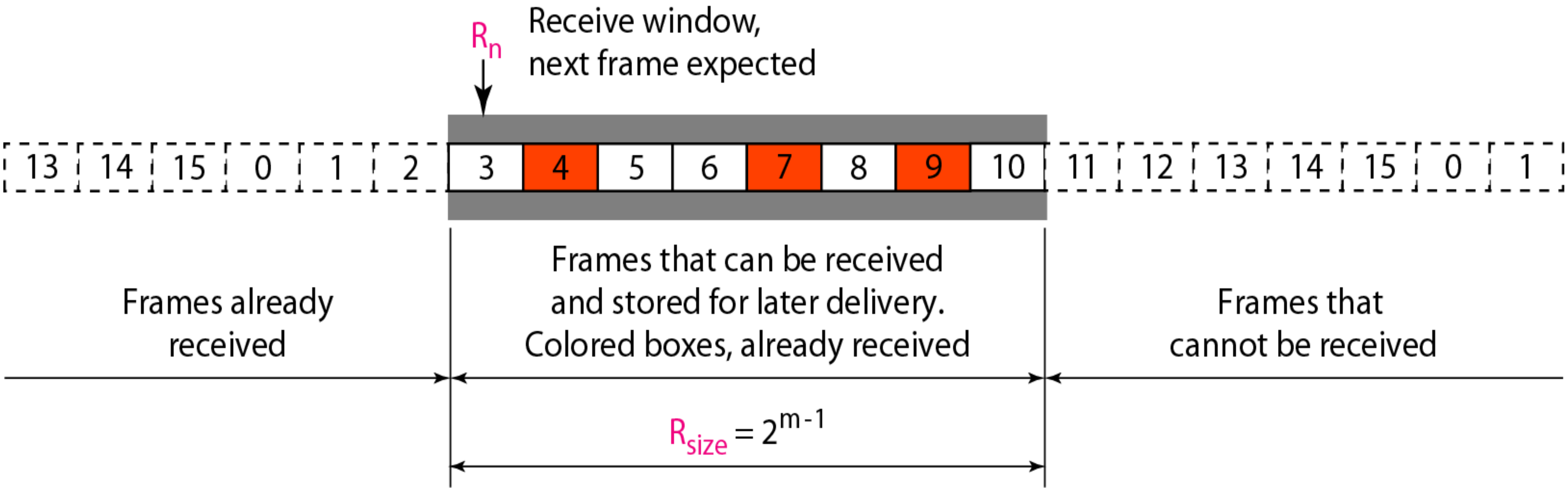
Frames sent, but not acknowledged

Frames that can be sent

Frames that cannot be sent

$S_{size} = 2^{m-1}$

Receive window for Selective Repeat ARQ



R_n Receive window, next frame expected

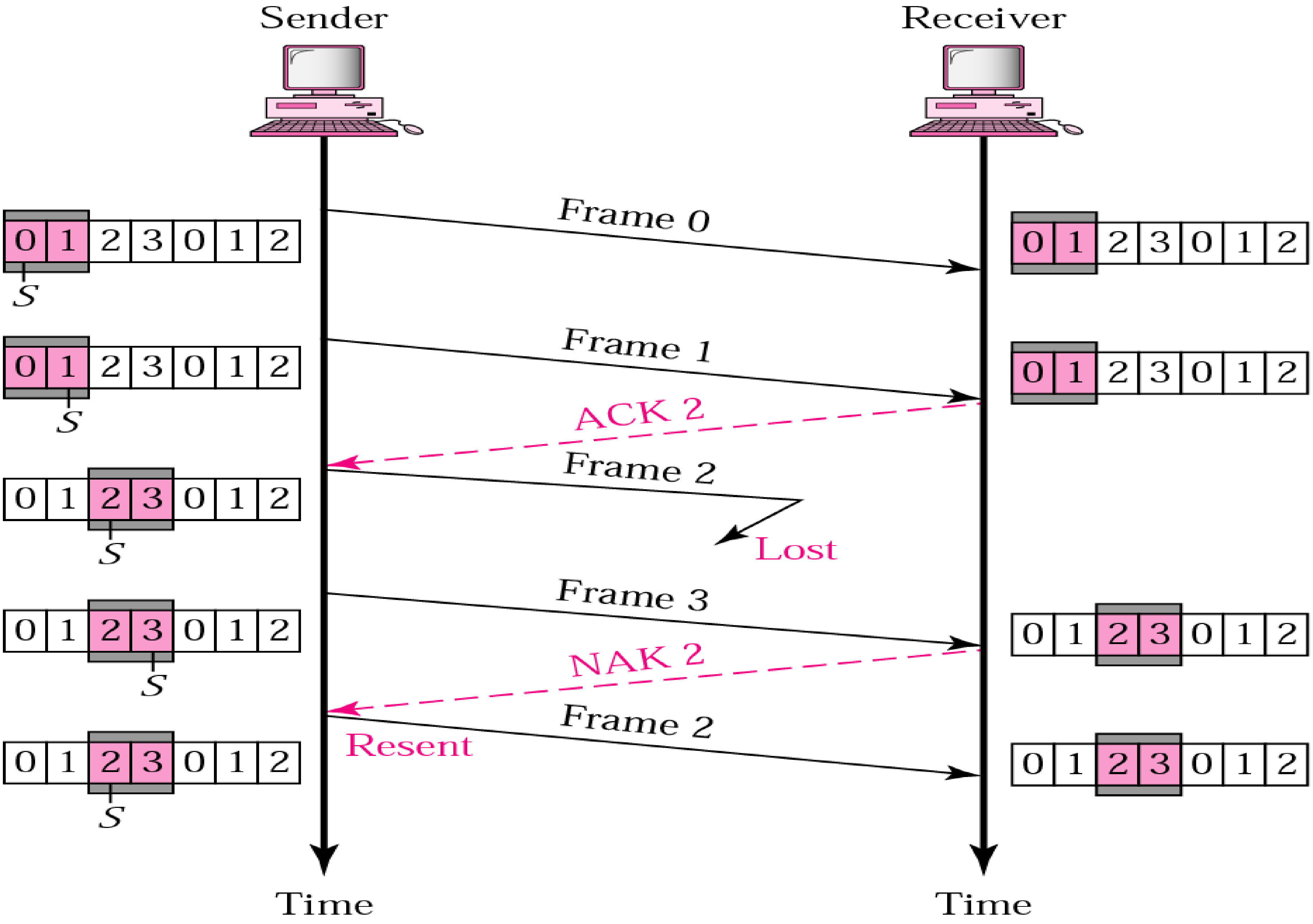
Frames already received

Frames that can be received and stored for later delivery. Colored boxes, already received

Frames that cannot be received

$R_{size} = 2^{m-1}$

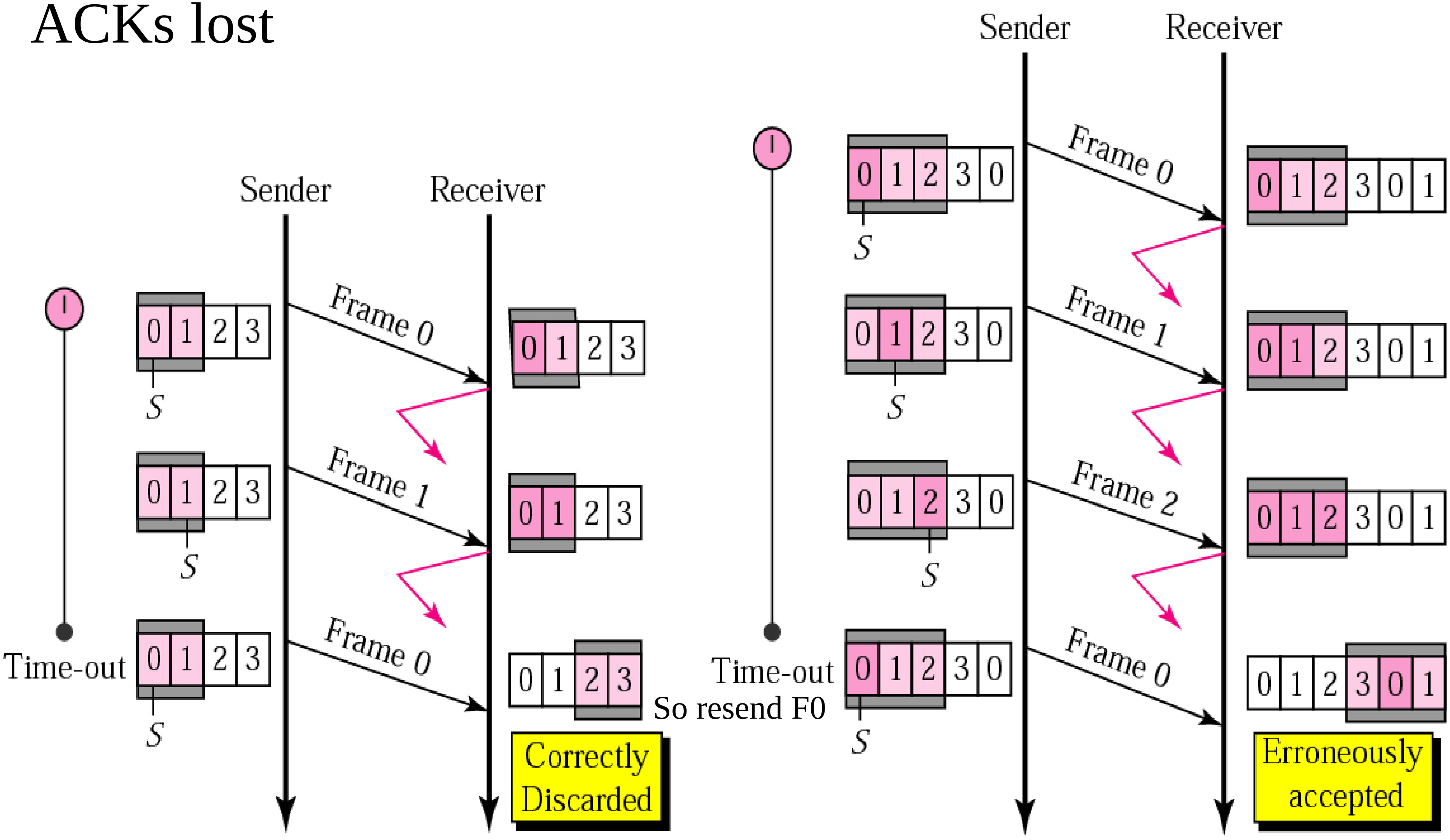
Selective Repeat ARQ, lost frame



Preferable sx and rx wnd size

- size of sx and rx wnds = at most one half of 2^m
- m = 2, size of the window is $2^m/2 = 2$

Selective Repeat ARQ, window size 2 and 3 and all ACKs lost



So resend F0

Correctly Discarded

Erroneously accepted

a. Window size = 2

b. Window size = 3

Note

In Selective Repeat ARQ, the size of the sender and receiver window must be at most one-half of 2^m.

Piggybacking

- protocols are unidirectional: data frames flow in only one direction although control information (ACK and NAK) can travel in the other direction.
- In real life, data frames are normally flowing in both directions: from A to B and vice versa
- piggybacking
  - When a frame is carrying data from A to B, it can also carry control information about arrived (or lost) frames from B;
  - when a frame is carrying data from B to A, it can also carry control information about the arrived (or lost) frames from A.

Thank You