# Selective-Repeat Protocol

- Go-Back-*N* protocol Each time a single packet is lost or corrupted, the sender resends all outstanding packets, to avoid this Another protocol, called the **Selective-Repeat (SR) protocol** is used resends only selective packets, those that are actually lost.

**Windows:**

- The Selective-Repeat protocol also uses two windows: a send window and a receive window. The send window maximum size can be $2^{m-1}$. The receive window in Selective-Repeat is totally different from the one in Go-Back-*N*. The size of the receive window is the same as the size of the send window(maximum $2m-1$).

- The Selective-Repeat protocol allows as many packets as the size of the receive window to arrive out of order and be kept until there is a set of consecutive packets to be delivered to the application layer.
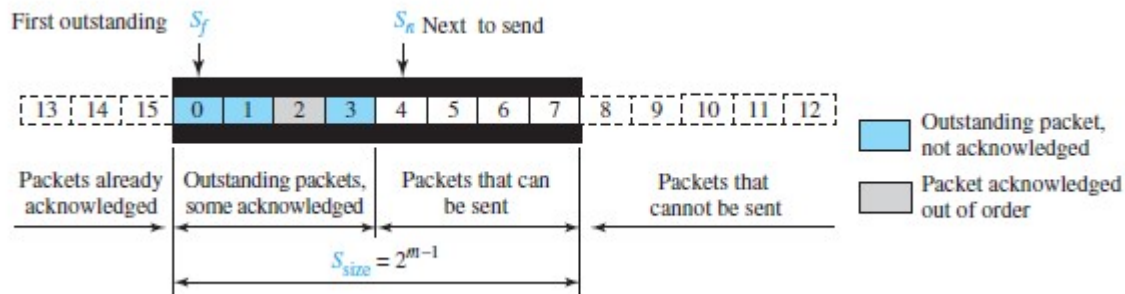


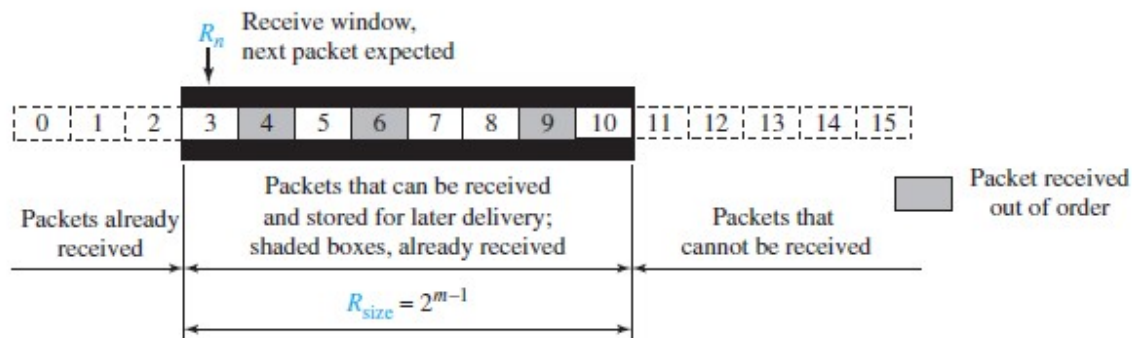Fig: Send window for Selective-Repeat protocol.



Fig: Receive window for Selective-Repeat protocol

**Timer:**

- Selective-Repeat uses one timer for each outstanding packet. When a timer expires, only the corresponding packet is resent.

**Acknowledgments:**

- In the Selective-Repeat protocol, an acknowledgment number defines the sequence number of the error-free packet received.

**FSMs:**

FSMs for the Selective-Repeat protocol.

*Sender:*

- The sender starts in the *ready* state, but later it can be in one of the two states: *ready* or *blocking*.

*Ready state.*

Four events may occur in this case:

a. If a request comes from the application layer, the sender creates a packet with the sequence number set to *Sn*. A copy of the packet is stored, and the packet is sent. If the timer is not running, the sender starts the timer. The value of *Sn* is now incremented, *Sn* = (*Sn* + 1) modulo 2*m*. If the window is full, *Sn* = (*Sf* +*S*size) modulo 2*m*, the sender goes to the blocking state.

b. If an error-free ACK, If there are outstanding packets, the timer is restarted; otherwise, the timer is stopped.

c. If a corrupted ACK or an error-free ACK with ackNo not related to an outstanding packet arrives, it is discarded.

d. If a time-out occurs, the sender resends all unacknowledged packets in the windowand restarts the timer.

*Blocking state.*

Three events may occur in this case:

- If an error-free ACK arrives with ackNo related to one of the outstanding packets, that packet is marked as acknowledged.

- If a corrupted ACK or an error-free ACK with the ackNo not related to outstanding packets arrives, the ACK is discarded.

- If a time-out occurs, the sender resends all unacknowledged packets in the window and restarts the timer.

*Receiver:*

The receiver is always in the *ready* state. Three events may occur:

a. If an error-free packet with seqNo in the window arrives, the packet is stored and an ACK with ackNo = seqNo is sent. In addition, if the seqNo = *Rn*, then the packet and all previously arrived consecutive packets are delivered to the application ayer and the window slides.

b. If an error-free packet with seqNo outside the window arrives, the packet is discarded, but an ACK with ackNo = *Rn* is returned to the sender.

c. If a corrupted packet arrives, the packet is discarded.

Sender

Time-out.
Resend all
outstanding packets
in window.
Reset the timer.

Request came from process.
Make a packet (seqNo = $S_n$).
Store a copy and send the packet.
Start the timer for this packet.
Set $S_n = S_n + 1$.

Window full
$(S_n = S_f + S_{size})$?

Time-out.
Resend all
outstanding packets
in window.
Reset the timer.

[false]      [true]

Start → Ready      [true]      Blocking

[false]

Window slides?

A corrupted ACK or
an ACK about a non-
outstanding packet
arrived.

Discard it.

A corrupted ACK or
an ACK about a non-
outstanding packet
arrived.

Discard it.

An error-free ACK arrived that
acknowledges one of the outstanding
packets.

Mark the corresponding packet.
If ackNo = $S_f$, slide the window over
all consecutive acknowledged packets.
If there are outstanding packets,
restart the timer. Otherwise, stop the
timer.

**Note:**
All arithmetic equations
are in modulo $2^m$.

Receiver

Error-free packet with seqNo
inside window arrived.

If duplicate, discard; otherwise,
store the packet.
Send an ACK with ackNo = seqNo.
If seqNo = $R_n$, deliver the packet and
all consecutive previously arrived
and stored packets to application,
and slide window.

**Note:**
All arithmetic equations
are in modulo $2^m$.

Ready

Corrupted packet arrived.

Discard the packet.

Start

Error-free packet with seqNo
outside window boundaries arrived.

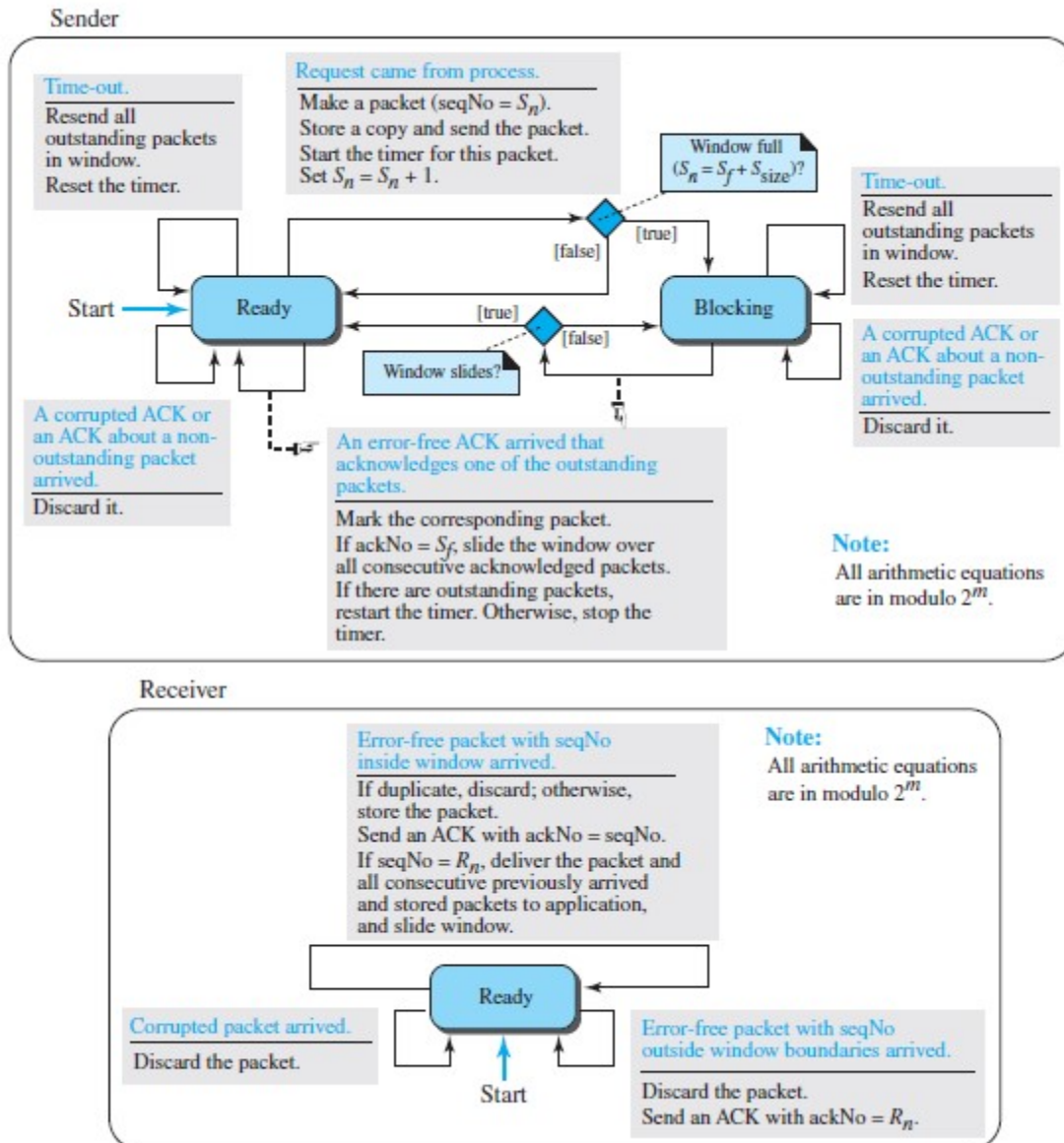Discard the packet.
Send an ACK with ackNo = $R_n$.

Fig: FSMs for SR protocol.

Example: 2

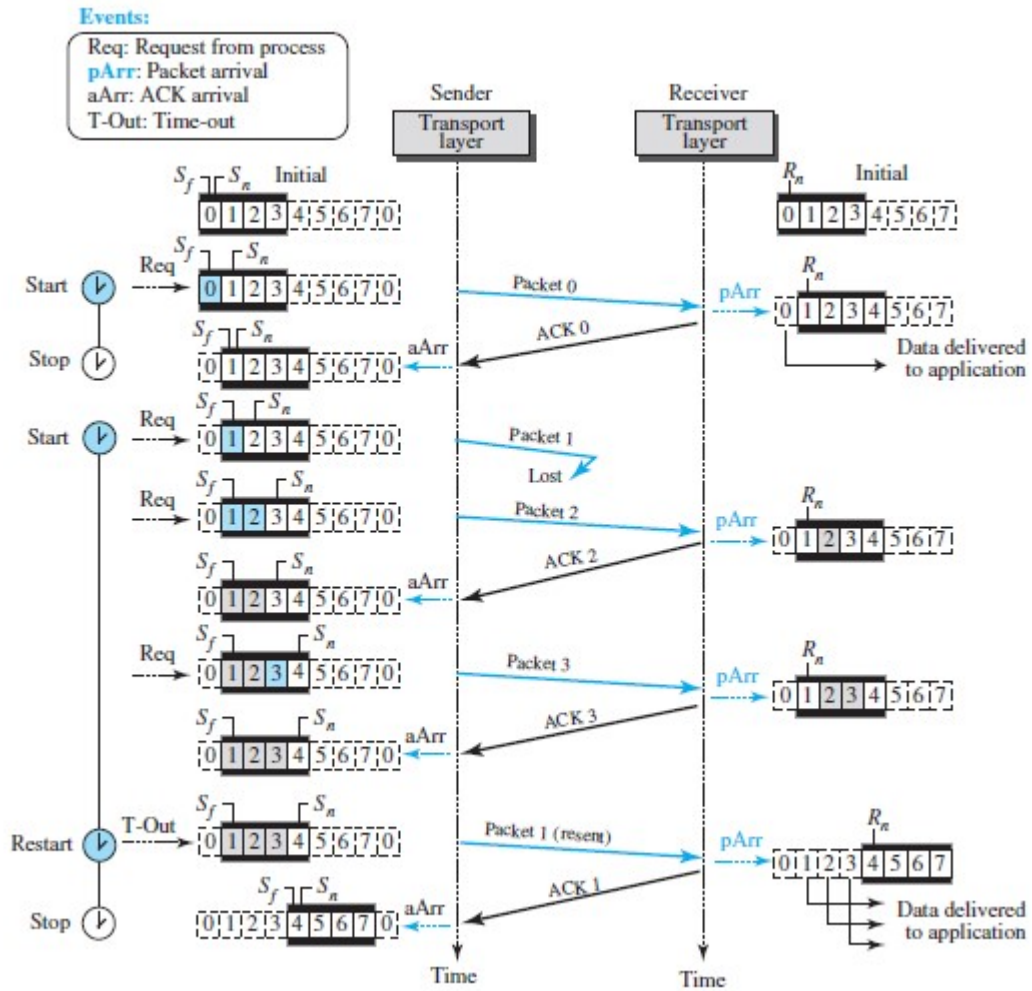This example show how Selective-Repeat behaves in this case. Fig  shows the situation.

Fig: Flow diagram for example 2