Data Link Control Layer (DLC)

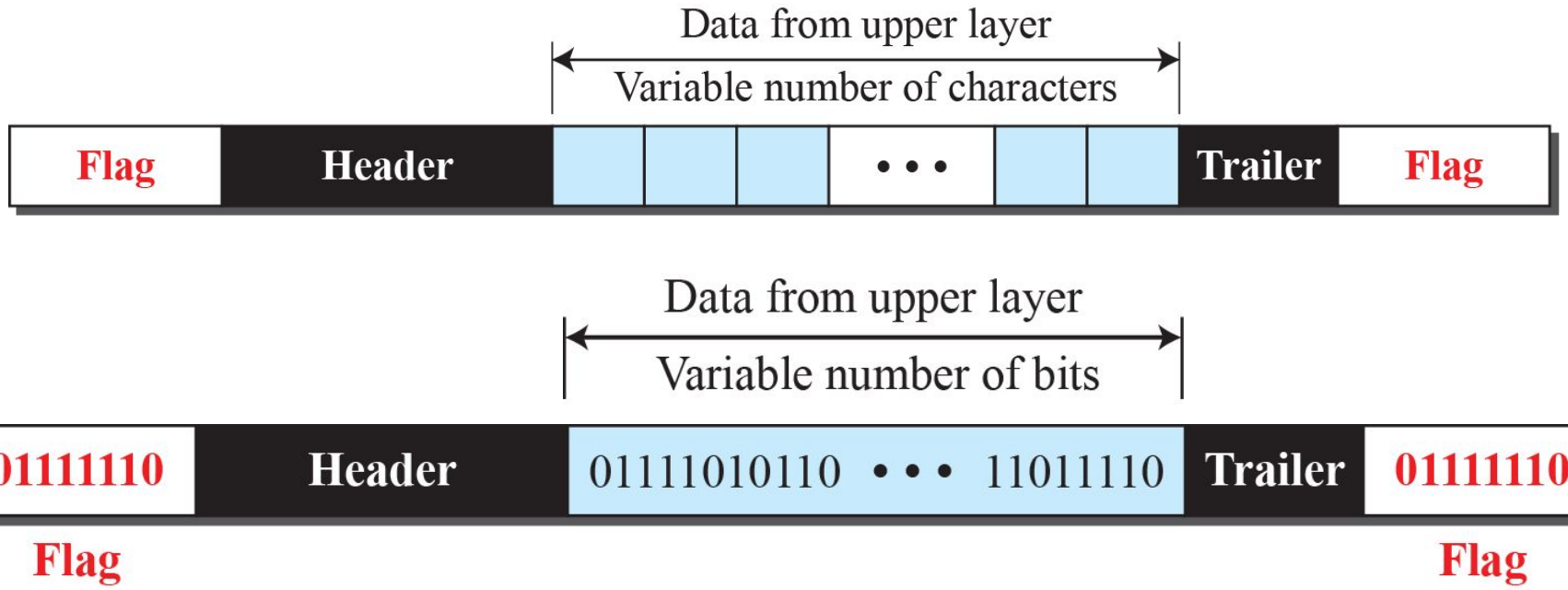
Session Objectives

After going through this session you will be able to understand

- ✔ Flow control
- ✔ Protocols for flow control

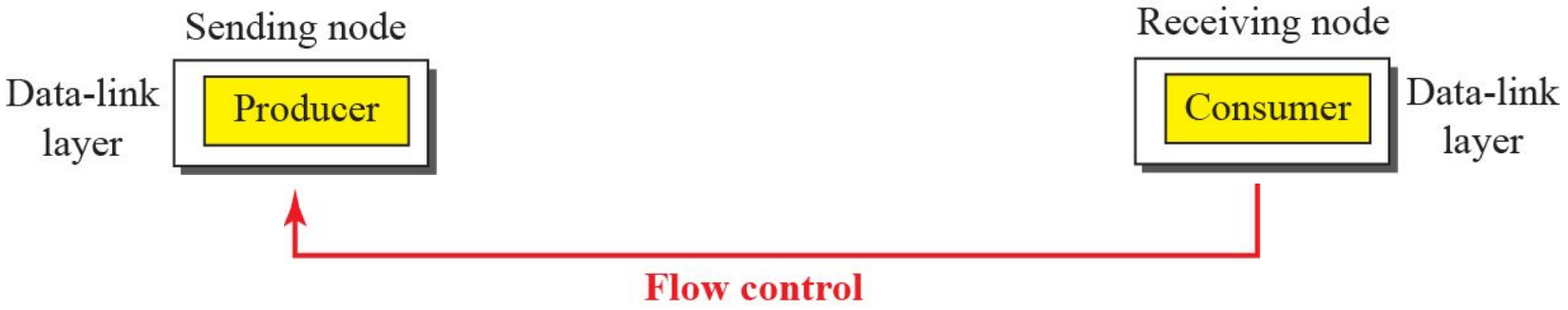
DLC Services

- The data link control (DLC) deals with procedures for communication between two adjacent nodes no matter whether the link is dedicated or broadcast.
- Data link control functions include framing, flow and error control.
- Our postal system practices a type of framing. The simple act of inserting a letter into an envelope separates one piece of information from another; the envelope serves as the delimiter.
- Framing in the data-link layer separates a message from one source to a destination by adding a sender address and a destination address. The destination address defines where the packet is to go; the sender address helps the recipient acknowledge the receipt.



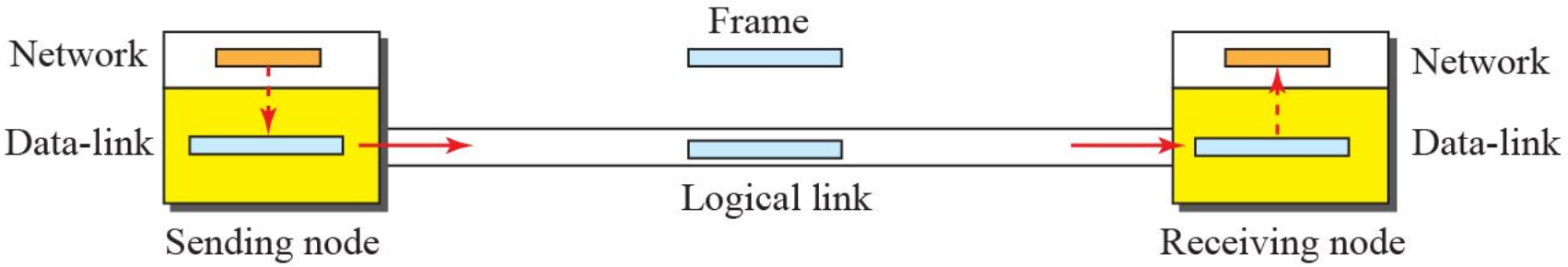
Flow and Error Control

- One of the responsibilities of the data-link control sublayer is flow and error control at the data-link layer.
- Traditionally four protocols have been defined for the data-link layer to deal with flow and error control: Simple, Stop-and-Wait, Go-Back-N, and Selective-Repeat.

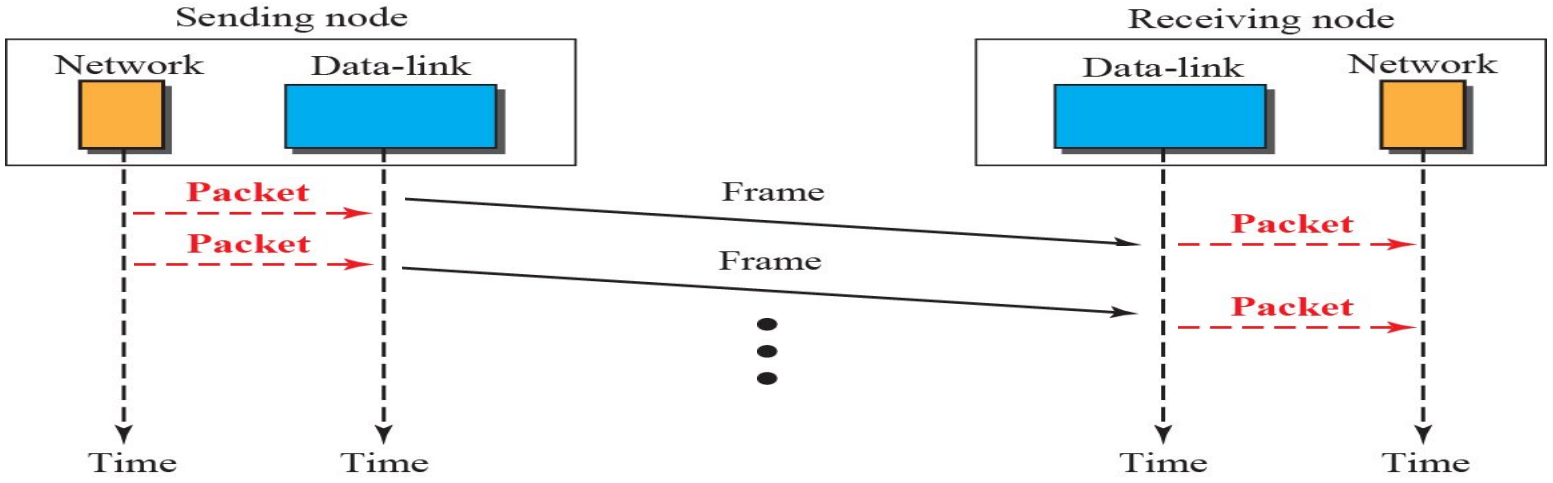


Simple Protocol

- Simple protocol has neither flow nor error control.
- It assumes that the receiver can immediately handle any frame it receives. In other words, the receiver can never be overwhelmed with incoming frames.

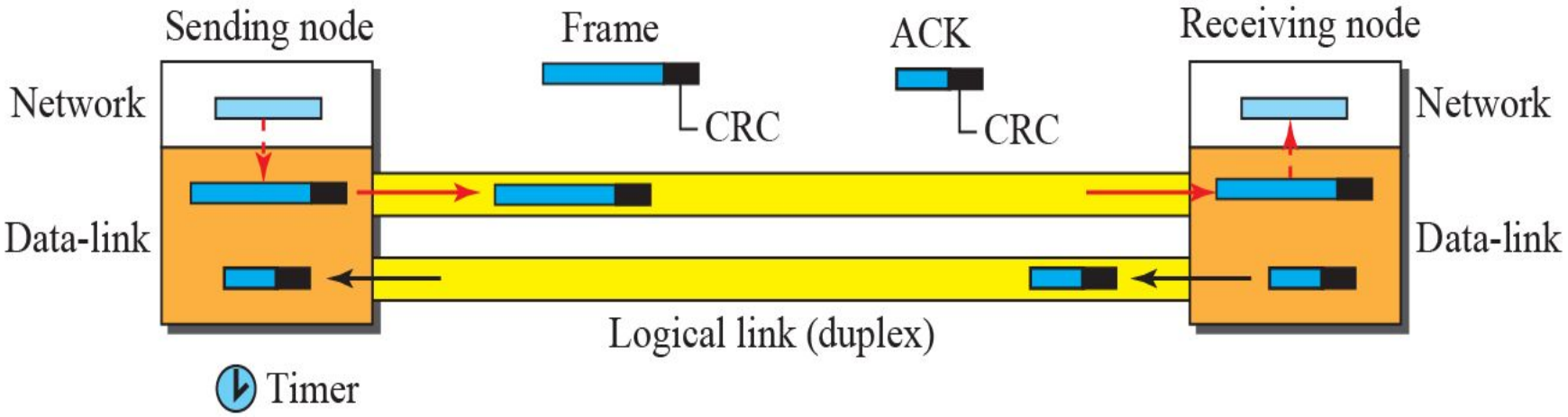


Example(Shown Below): The sender sends frames one after another without even thinking about the receiver.

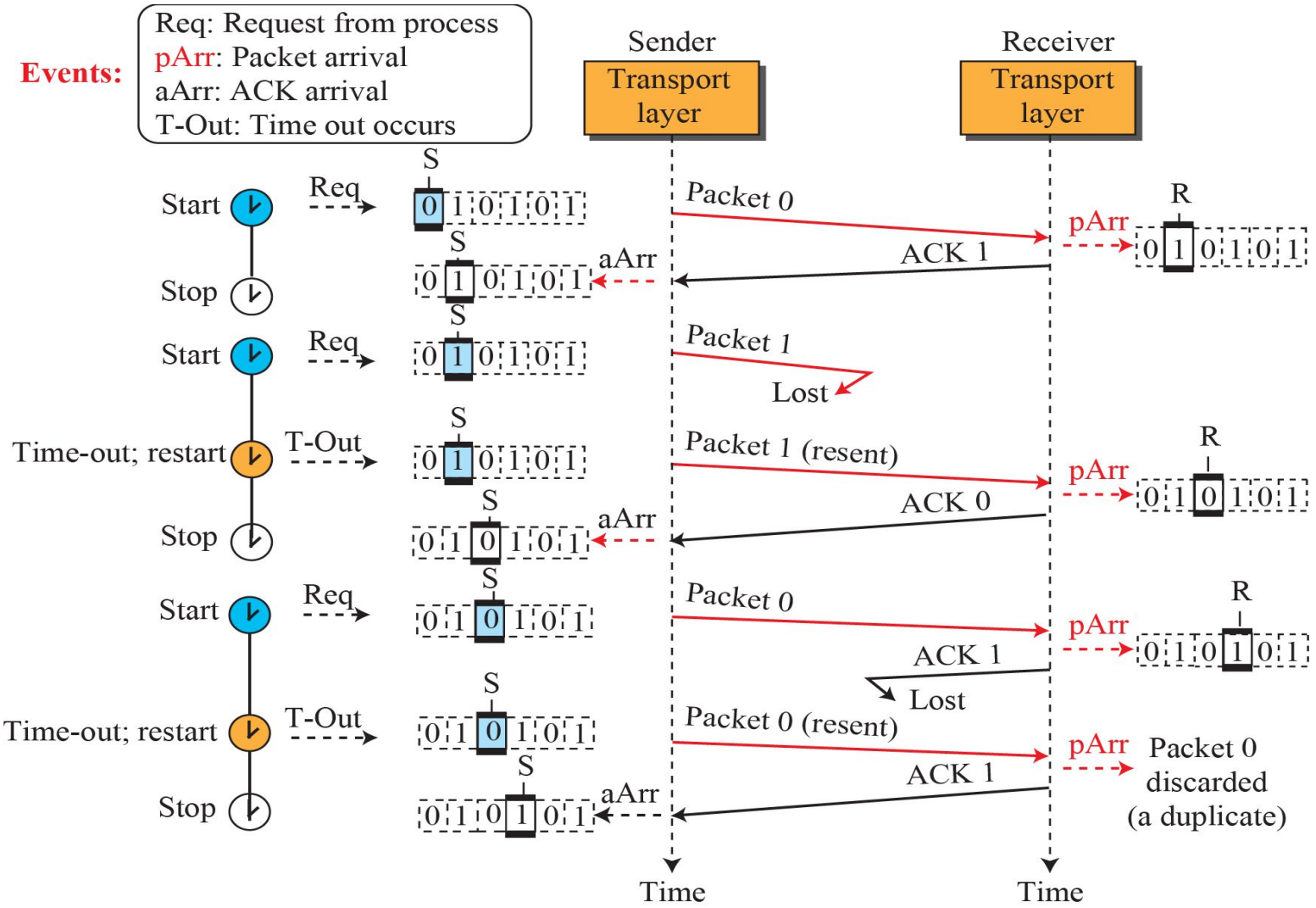


Stop and Wait Protocol

- The sender sends one frame at a time and waits for an acknowledgment before sending the next one.
- To detect corrupted frames, it need to add a CRC to each data frame

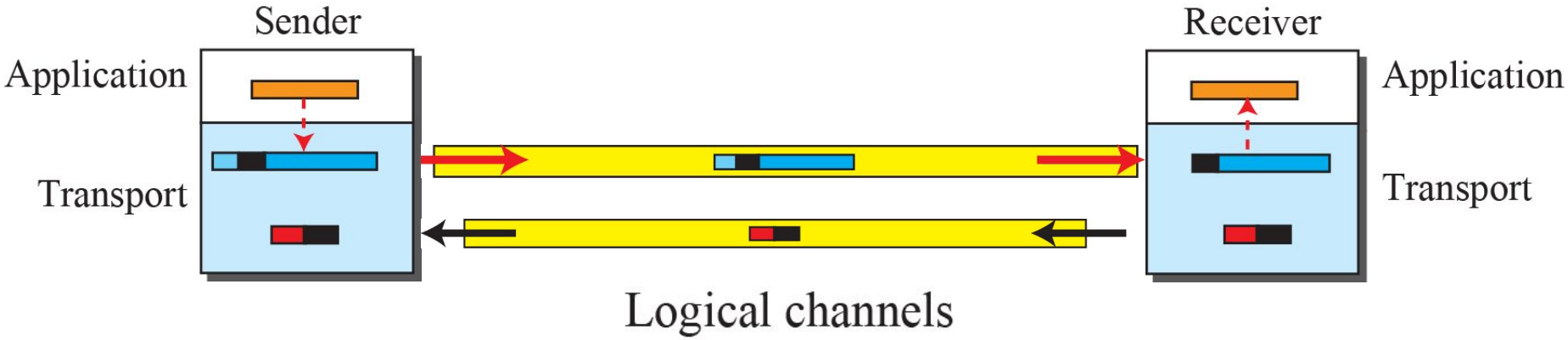


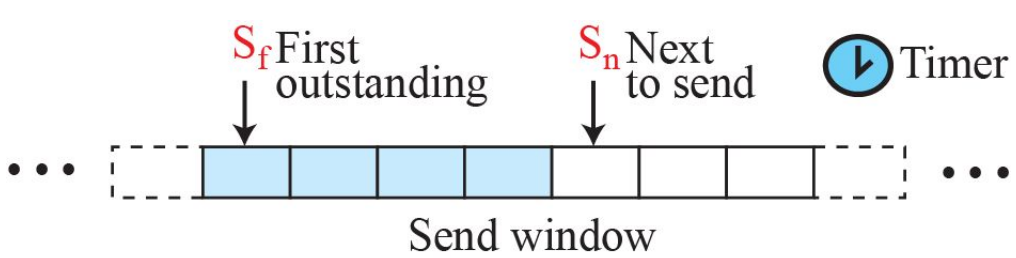
Example (Shown Below): In the Stop-and-Wait protocol. Packet 0 is sent and acknowledged. Packet 1 is lost and resent after the time-out. The resent packet 1 is acknowledged and the timer stops. Packet 0 is sent and acknowledged, but the acknowledgment is lost. The sender has no idea if the packet or the acknowledgment is lost, so after the time-out, it resends packet 0, which is acknowledged.

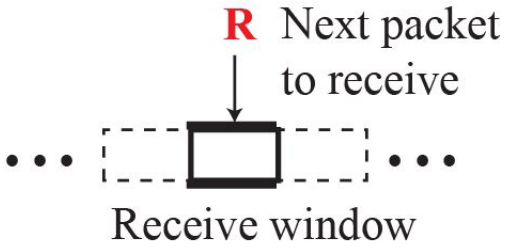


Go – Back -N Protocol

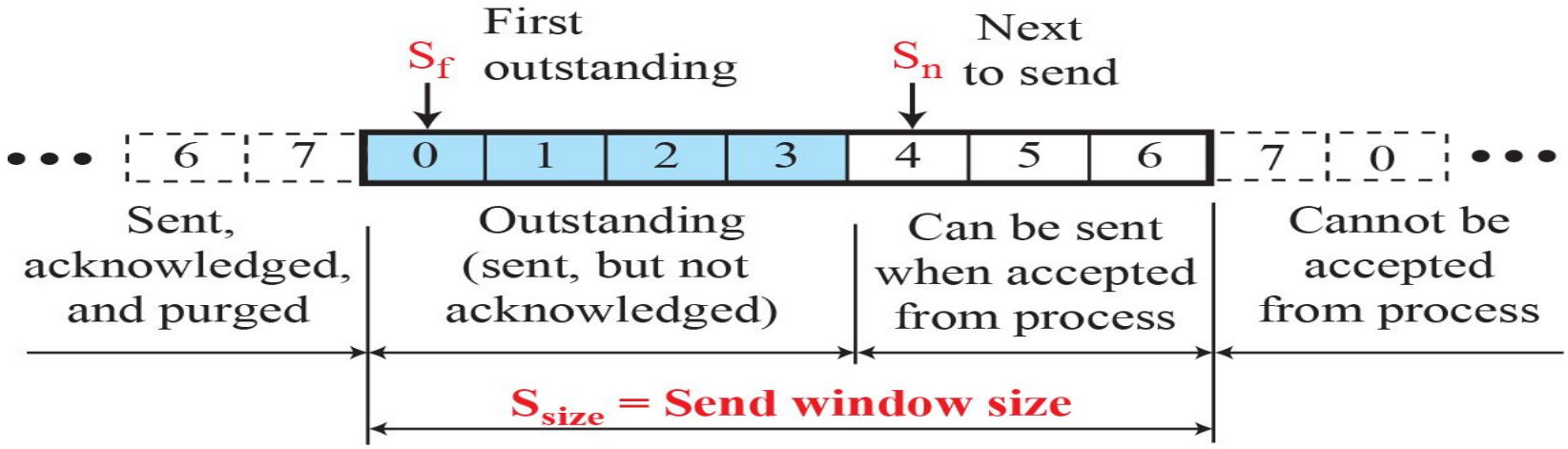
- To improve the efficiency of transmission (to fill the pipe), multiple packets must be in transition while the sender is waiting for acknowledgment.
- In other words, we need to let more than one packet be outstanding to keep the channel busy while the sender is waiting for acknowledgment.



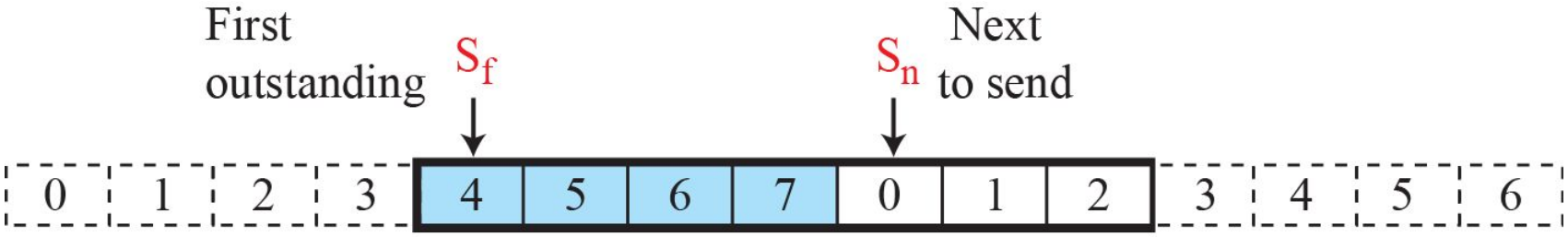




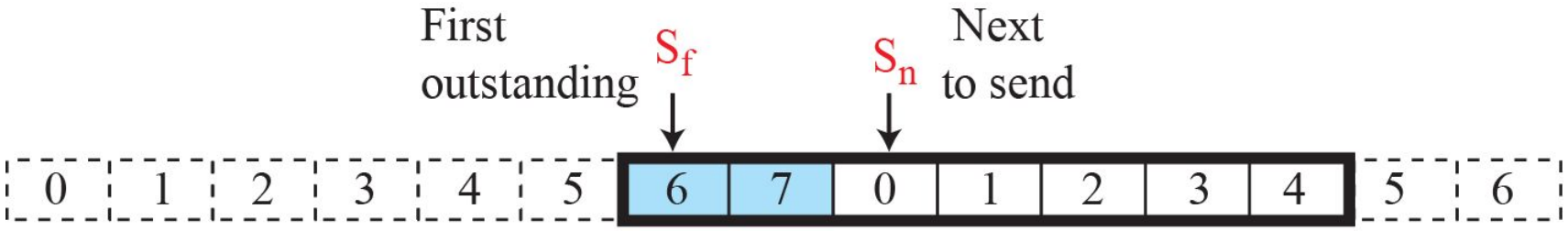
Send window for Go-Back-N



Sliding the send window

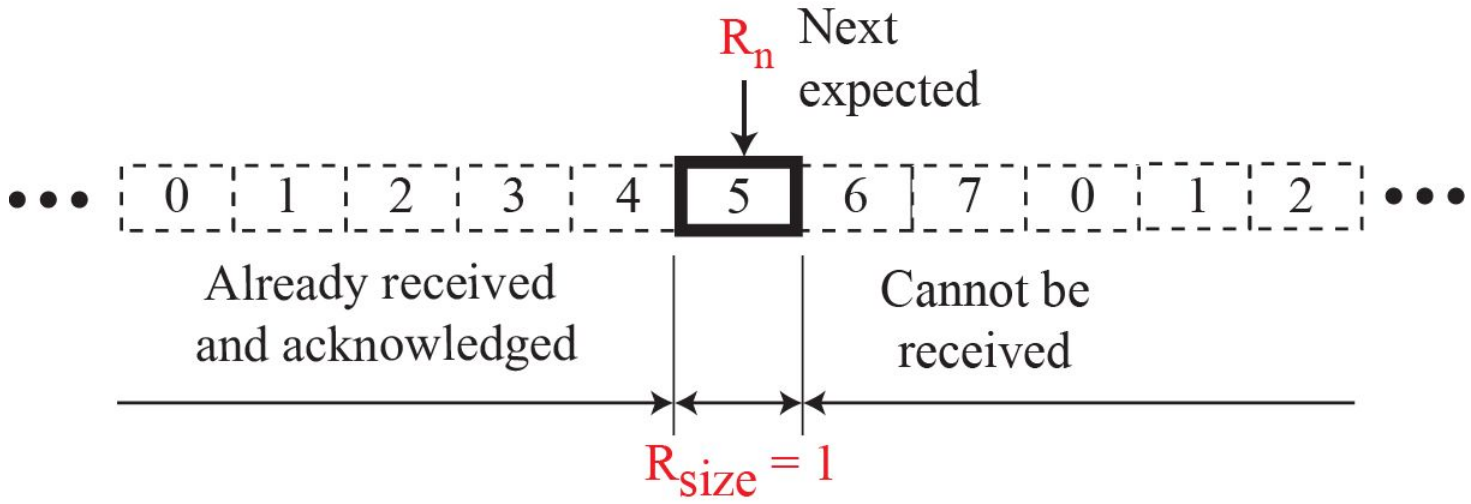


a. Window before sliding

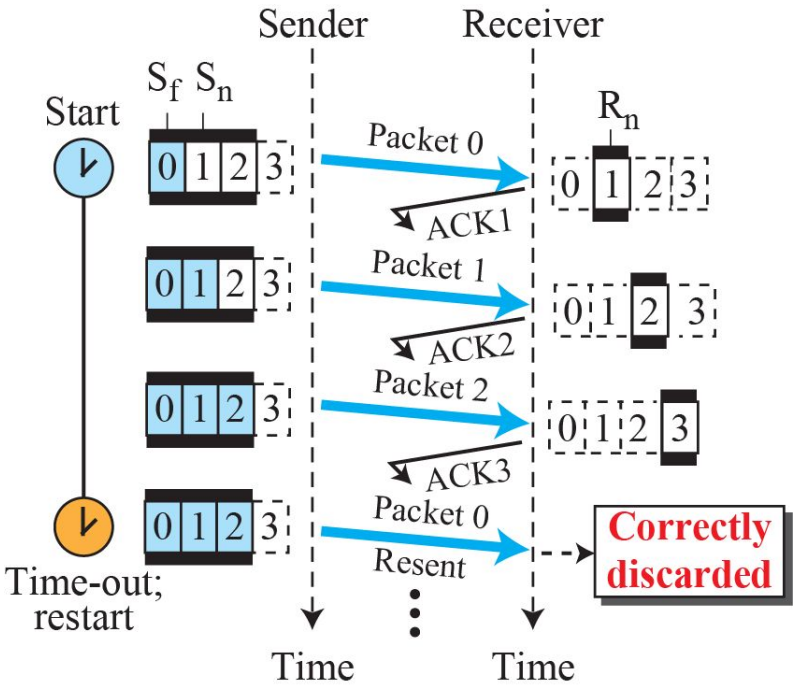


b. Window after sliding (an ACK with ackNo = 6 has arrived)

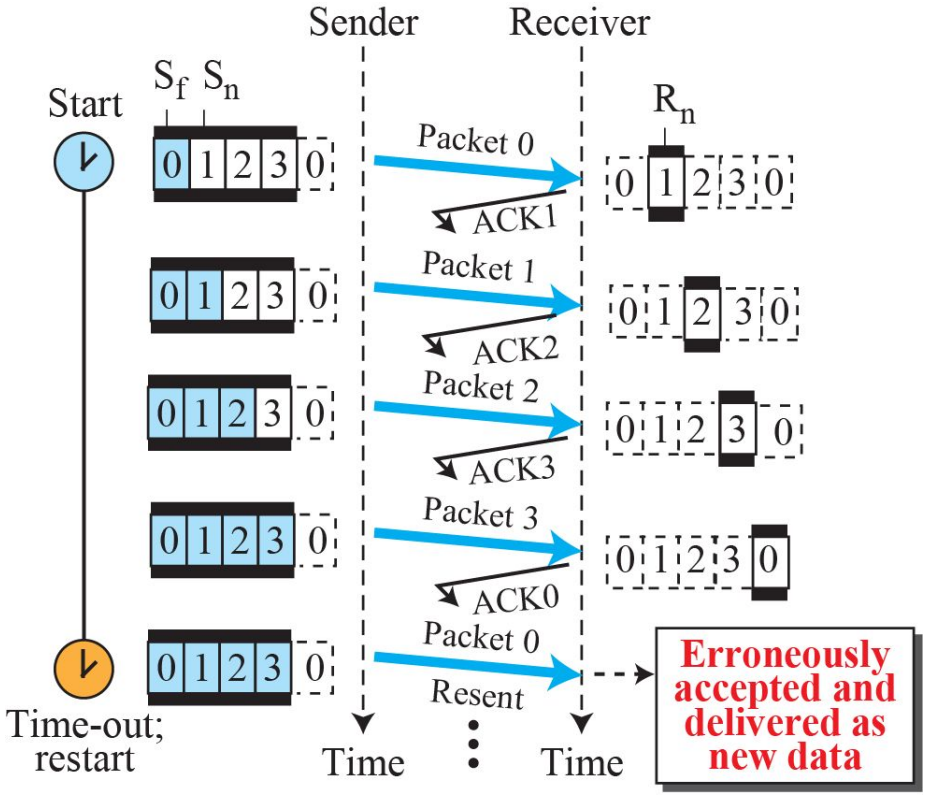
Receive window for Go-Back-N



Send window size for Go-Back-N

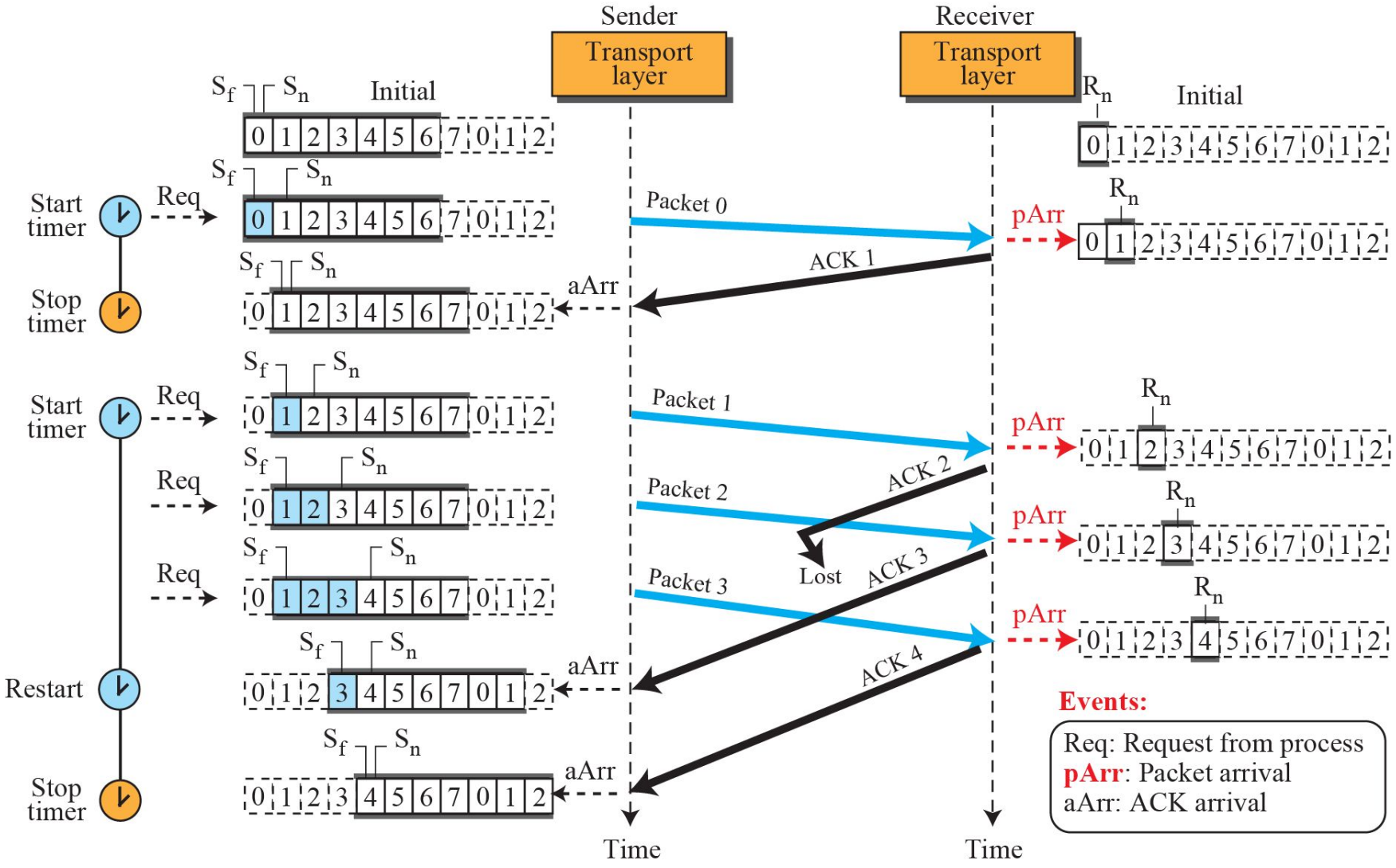


a. Send window of size < 2^m

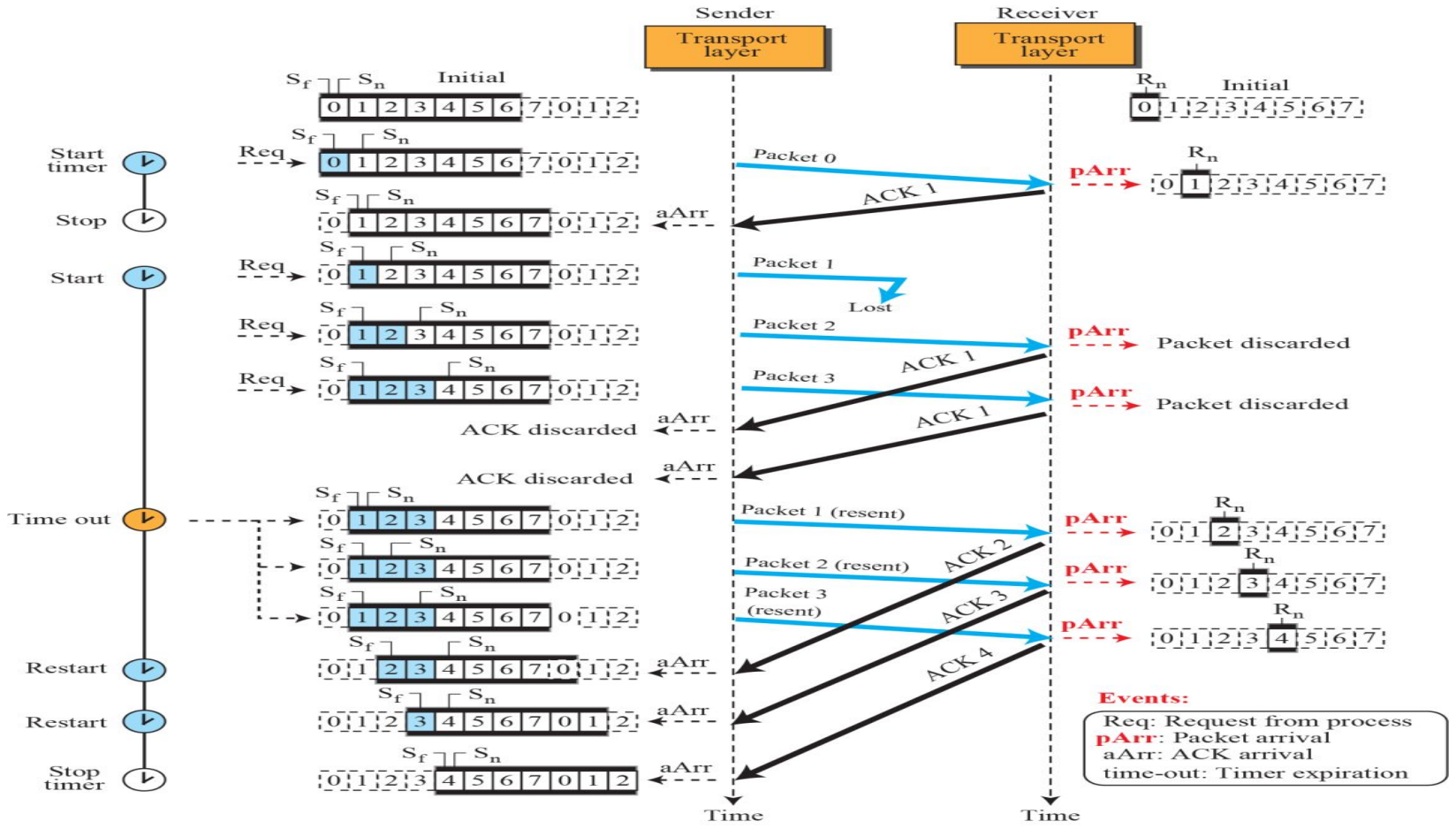


b. Send window of size = 2^m

Example 1(* Go-Back-*N): This is an example of a case where the forward channel is reliable, but the reverse is not. No data packets are lost, but some ACKs are delayed and one is lost. The example also shows how cumulative acknowledgments can help if acknowledgments are delayed or lost.

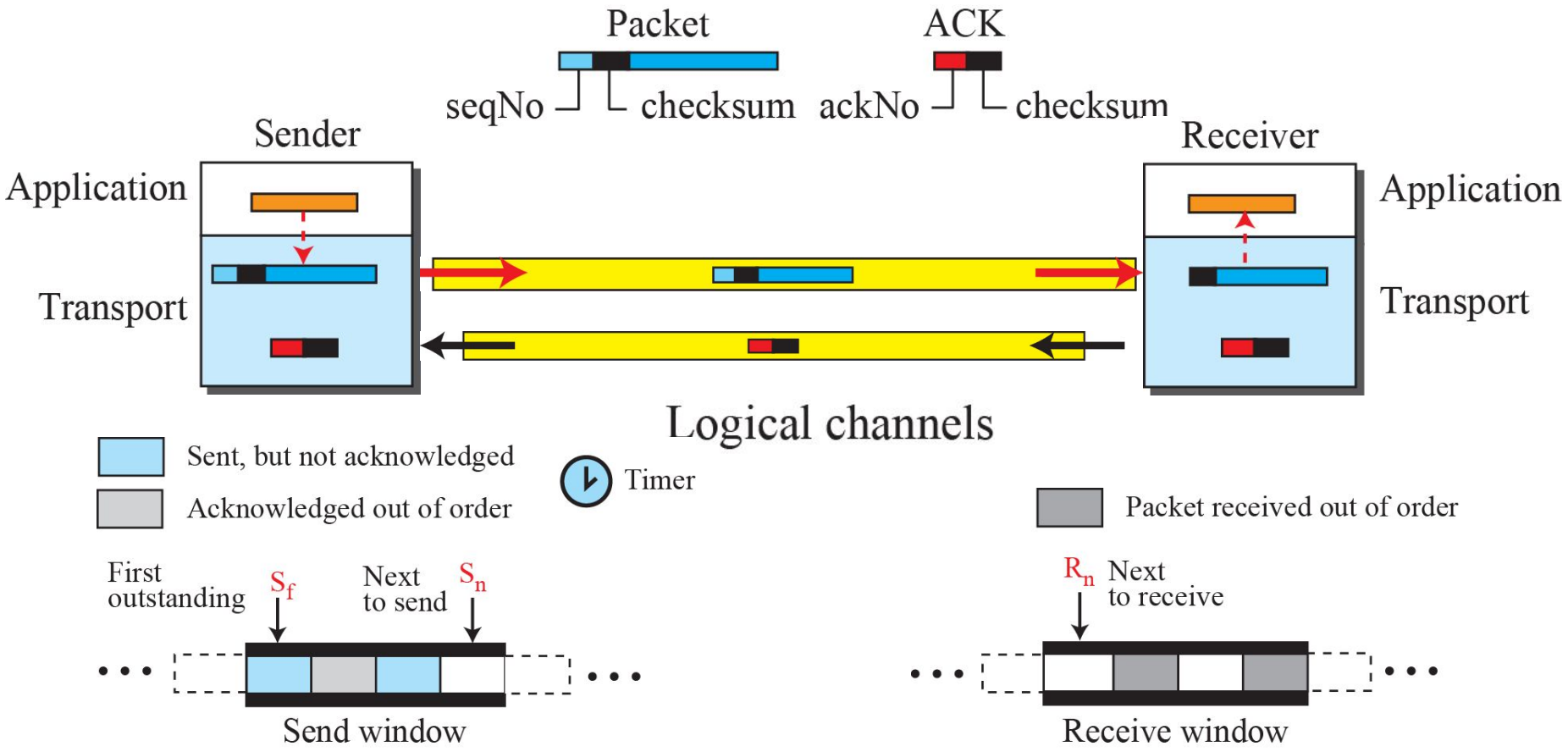


Example 2(* Go-Back-*N): what happens when a packet is lost. Packets 0, 1, 2, and 3 are sent. However, packet 1 is lost. The receiver receives packets 2 and 3, but they are discarded because they are received out of order (packet 1 is expected). When the receiver receives packets 2 and 3, it sends ACK1 to show that it expects to receive packet 2,3. However, these ACKs are not useful for the sender because the ack No is equal to S_f, not greater that S_f. So the sender discards them. When the time-out occurs, the sender resends packets 1, 2, and 3, which are acknowledged.

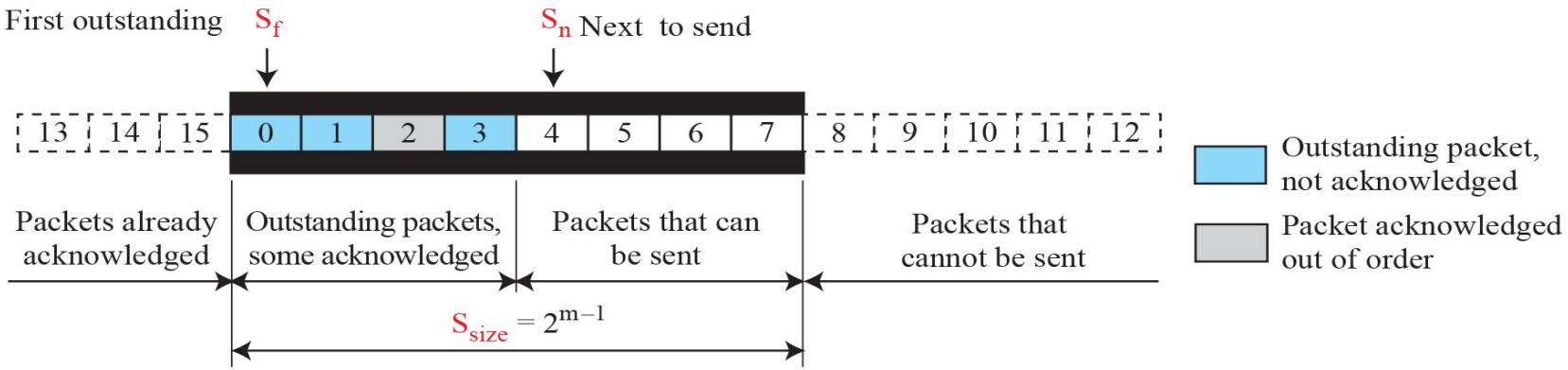


Selective Repeat Protocol

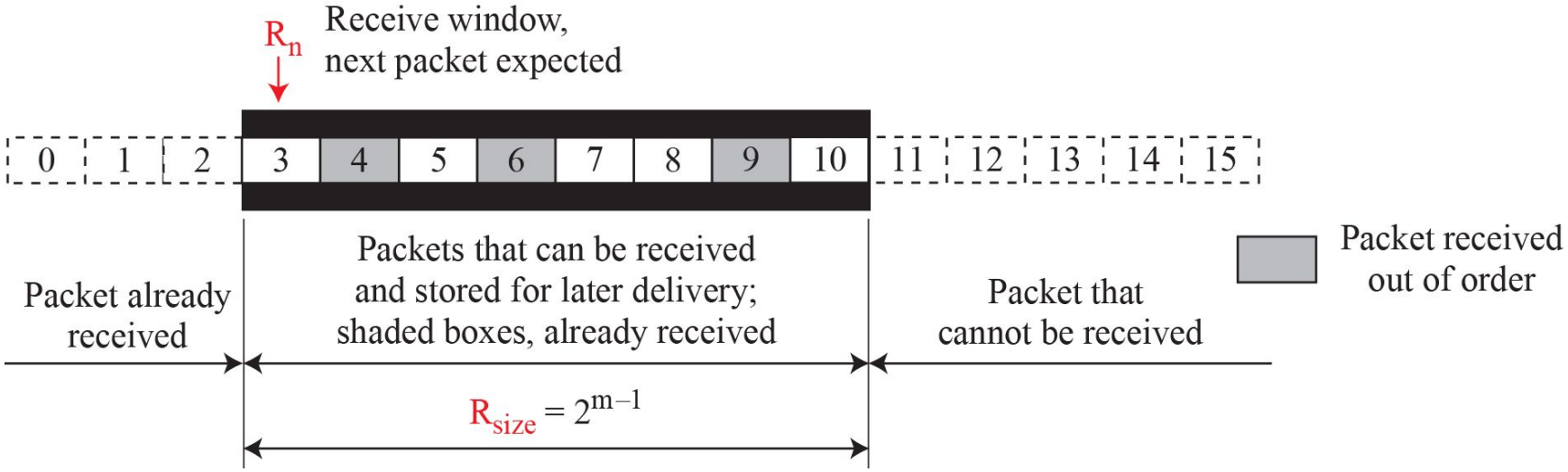
The Go-Back-N protocol simplifies the process at the receiver. The receiver keeps track of only one variable, and there is no need to buffer out-of-order packets; they are simply discarded. However, this protocol is inefficient if the underlying network protocol loses a lot of packets. Each time a single packet is lost or corrupted, the sender resends all outstanding packets, even though some of these packets may have been received safe and sound but out of order.



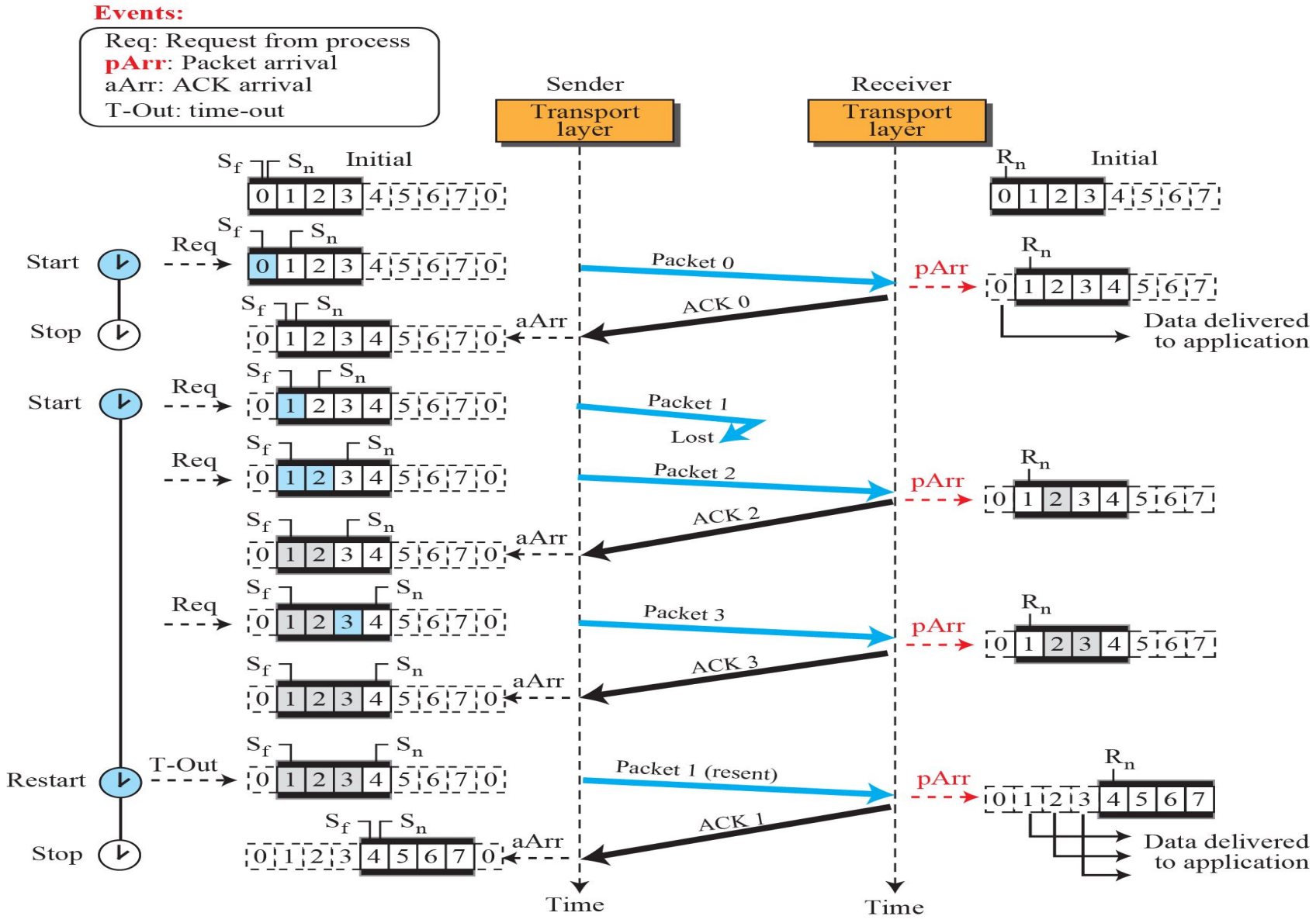
Send window for Selective-Repeat protocol



Receive window for Selective-Repeat protocol



Example 1(Selective Repeat): Flow diagram



Summary

In this section we have discussed the following:

- ✔ Flow Control
- ✔ Algorithm for flow control


Thank
you!