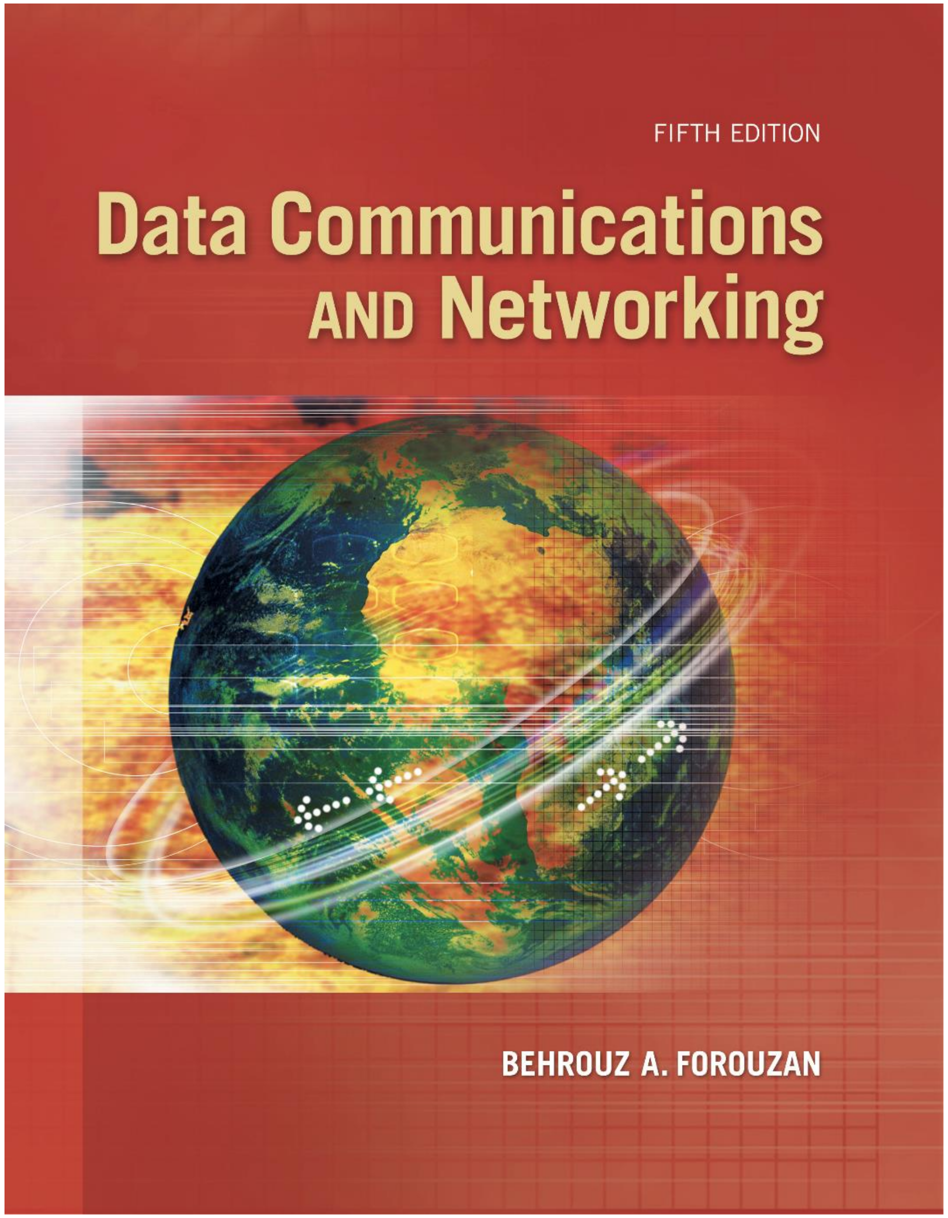
The McGraw-Hill Companies

Chapter 11

Data Link Control (DLC)



Chapter 11: Outline

11.1 DLC SERVICES

11.2 DATA-LINK LAYER PROTOCOLS

11-1 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 and flow and error control.

11.1.1 Framing

The data-link layer needs to pack bits into frames, so that each frame is distinguishable from another.

- *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.*

Figure 11.1: A frame in a character-oriented protocol

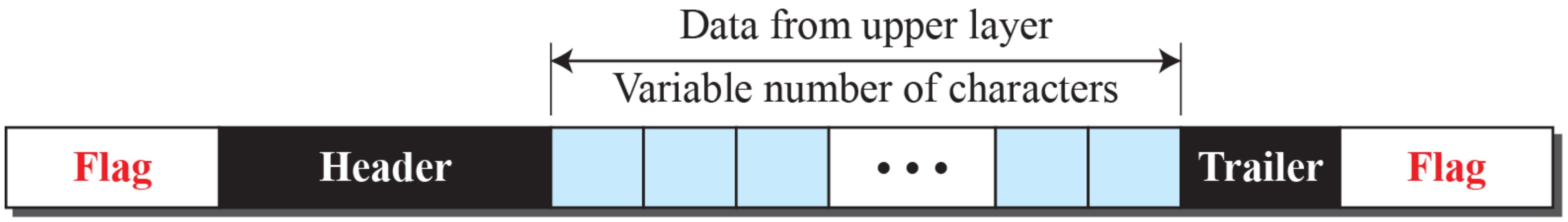


Figure 11.2: Byte stuffing and unstuffing

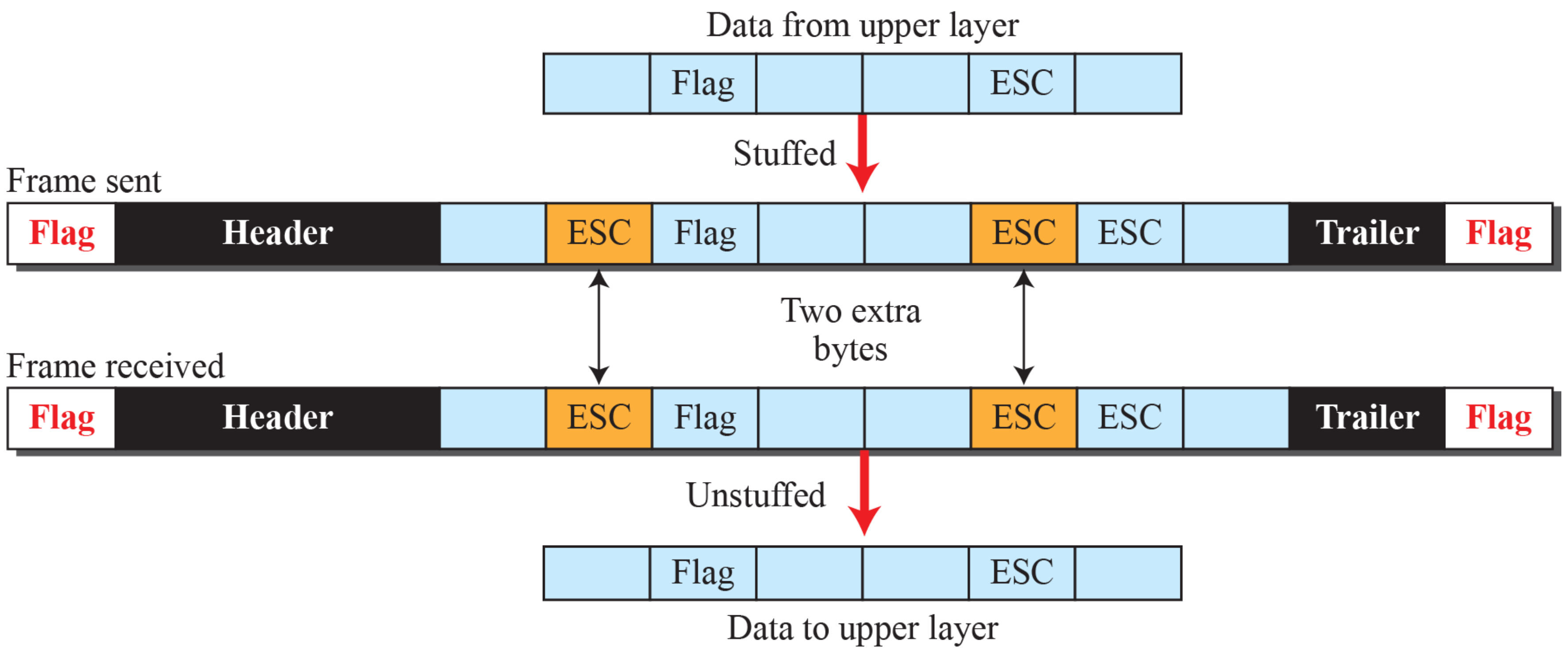


Figure 11.3: A frame in a bit-oriented protocol

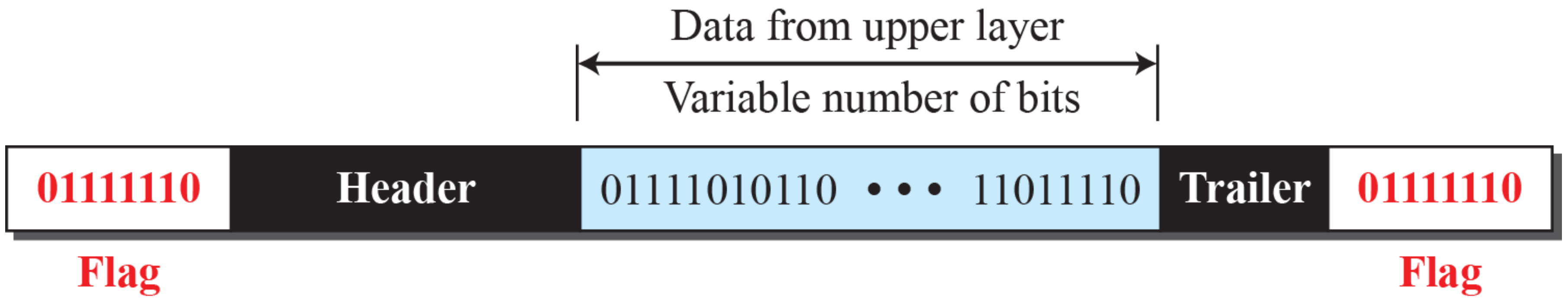
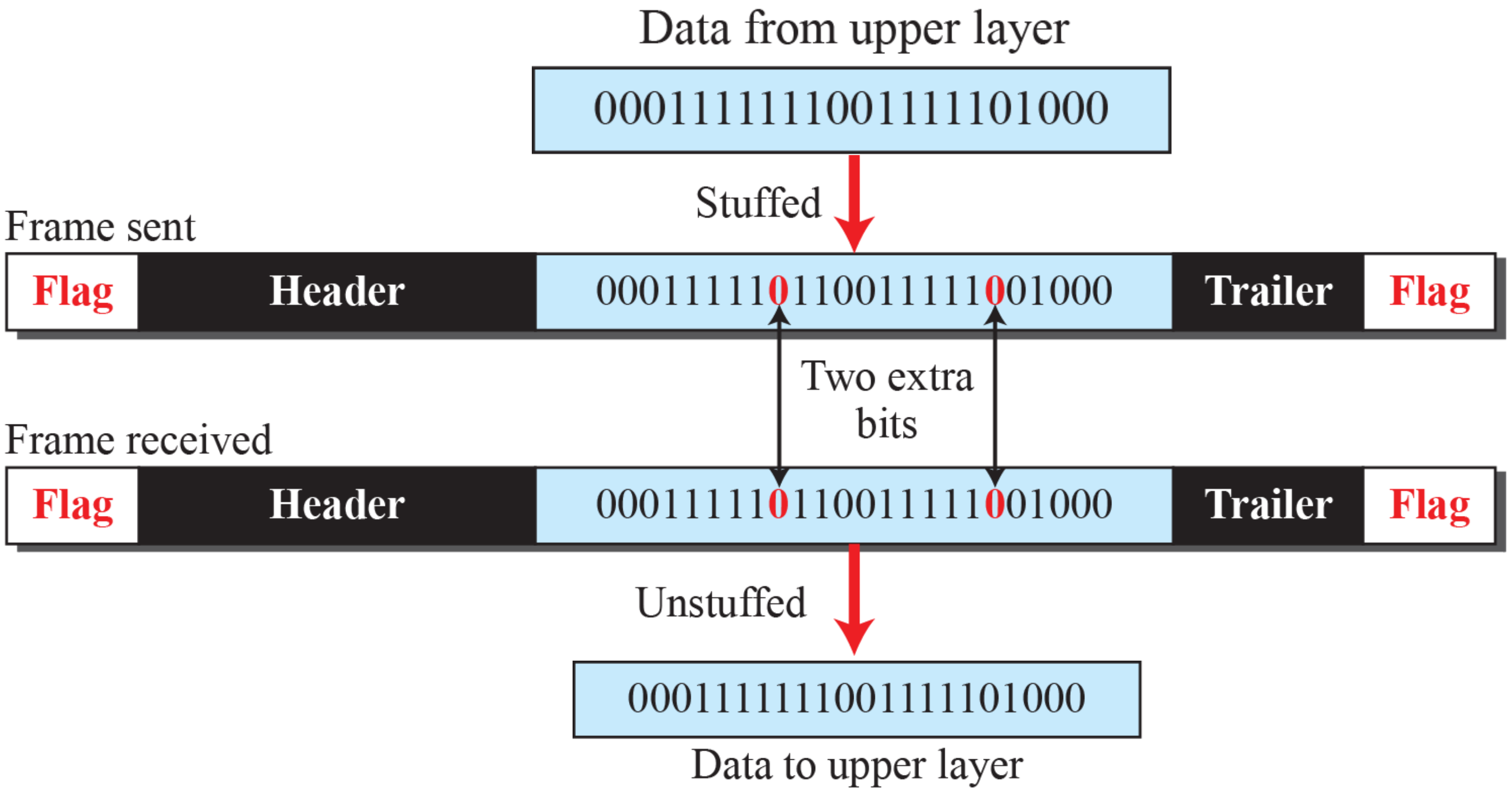


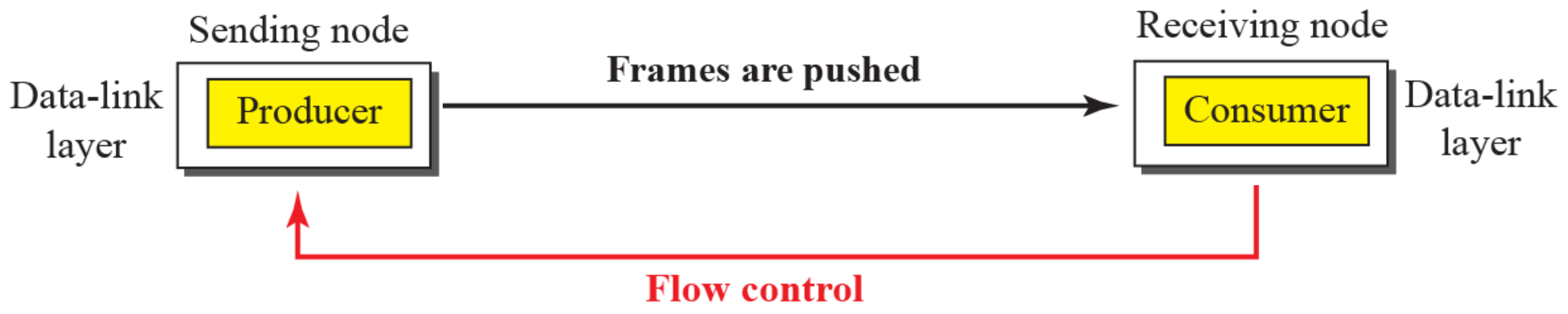
Figure 11.4: Bit stuffing and unstuffing



11.1.2 Flow and Error Control

One of the responsibilities of the data-link control sublayer is flow and error control at the data-link layer.

Figure 11.5: Flow control at the data link layer



Example 11.1

The above discussion requires that the consumers communicate with the producers on two occasions: when the buffer is full and when there are vacancies.

If the two parties use a buffer with only one slot, the communication can be easier. Assume that each data-link layer uses one single memory slot to hold a frame. When this single slot in the receiving data-link layer is empty, it sends a note to the network layer to send the next packet for framing.

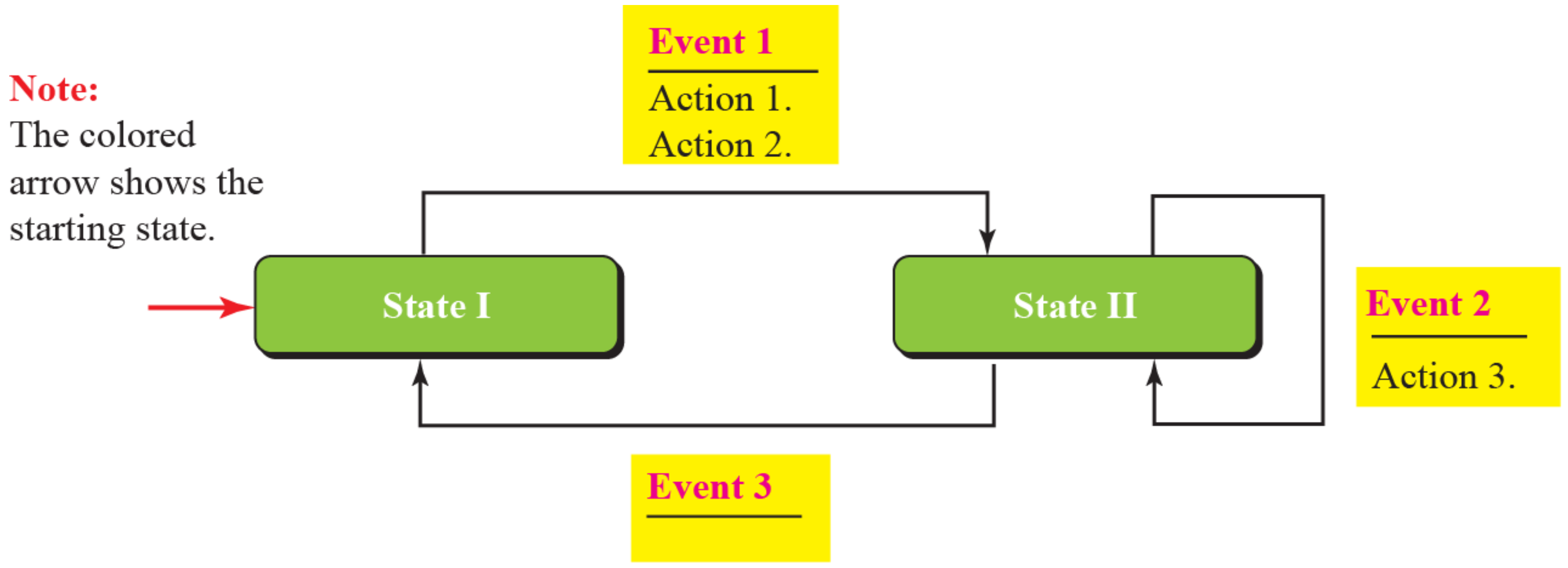
11.1.3 Connection

A DLC protocol can be either connectionless or connection-oriented.

11-2 DATA-LINK LAYER PROTOCOLS

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. Although the first two protocols still are used at the data-link layer, the last two have disappeared. We therefore briefly discuss the first two protocols.

Figure 11.6: FSMs



11.2.1 Simple Protocol

Our first protocol is a simple protocol with neither flow nor error control. We assume that the receiver can immediately handle any frame it receives. In other words, the receiver can never be overwhelmed with incoming frames. Figure 11.7 shows the layout for this protocol.

Figure 11.7: Simple protocol

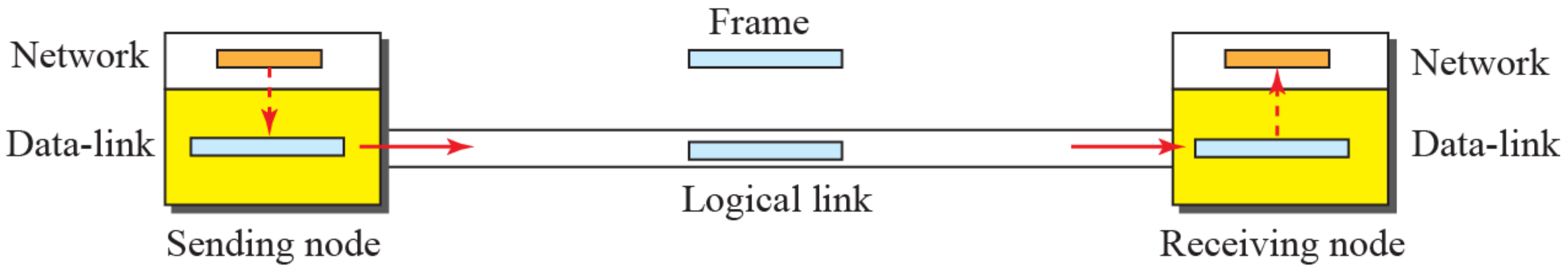
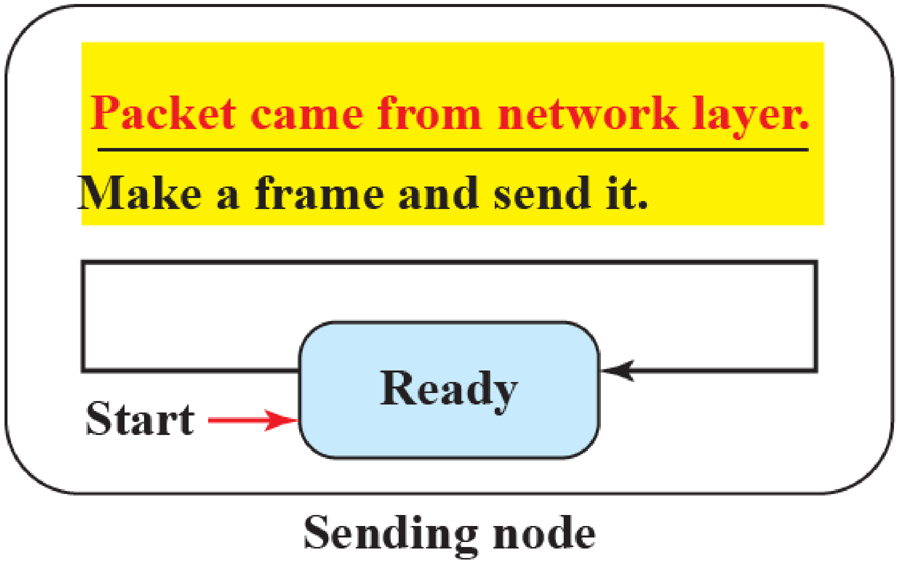
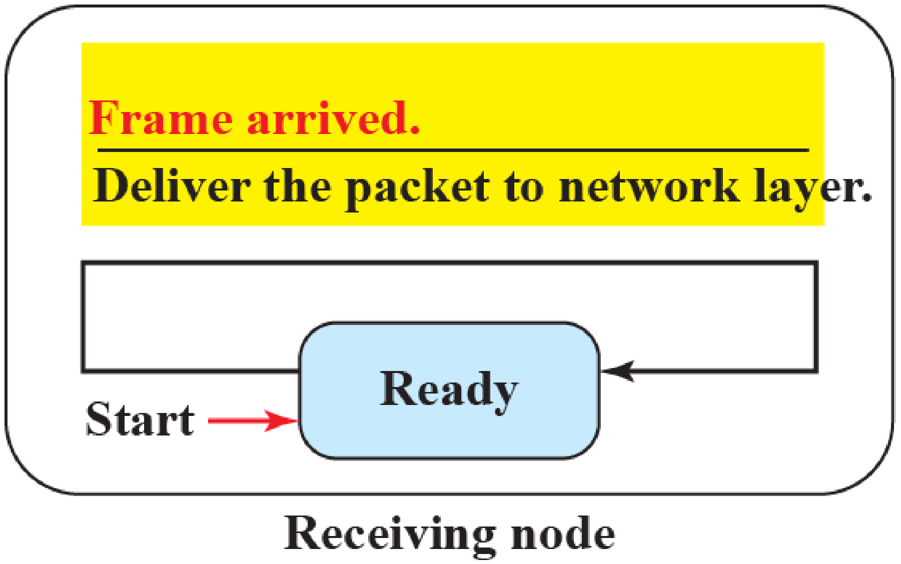


Figure 11.8: FSM for the simple protocol

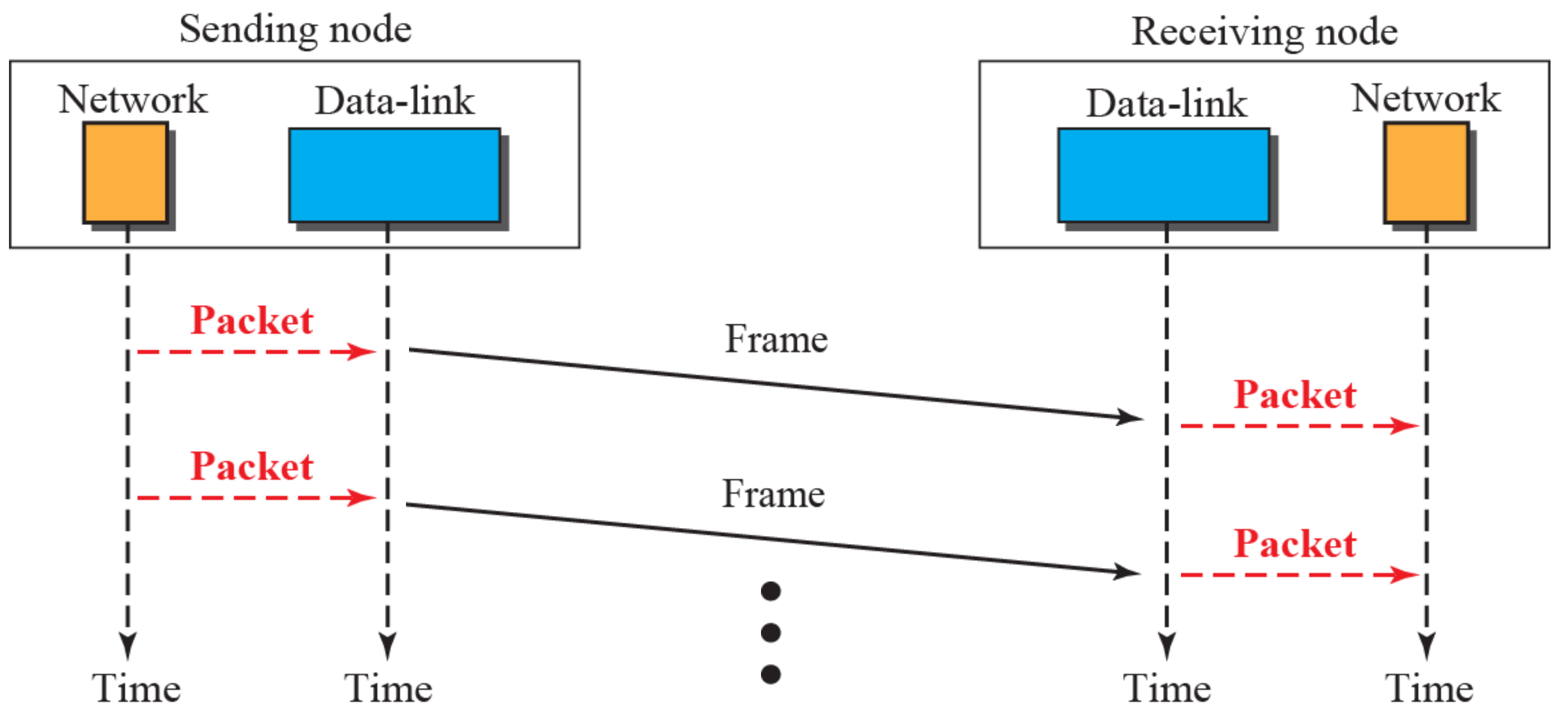




Example 11.2

Figure 11.9 shows an example of communication using this protocol. It is very simple. The sender sends frames one after another without even thinking about the receiver.

Figure 11.9: Flow diagram for Example 111.2



11.2.2 Stop-and-Wait Protocol

Our second protocol is called the Stop-and-Wait protocol, which uses both flow and error control. We show a primitive version of this protocol here, but we discuss the more sophisticated version in Chapter 23 when we have learned about sliding windows. In this protocol, the sender sends one frame at a time and waits for an acknowledgment before sending the next one. To detect corrupted frames, we need to add a CRC to each data frame.

Figure 11.10: Stop-and-wait Protocol

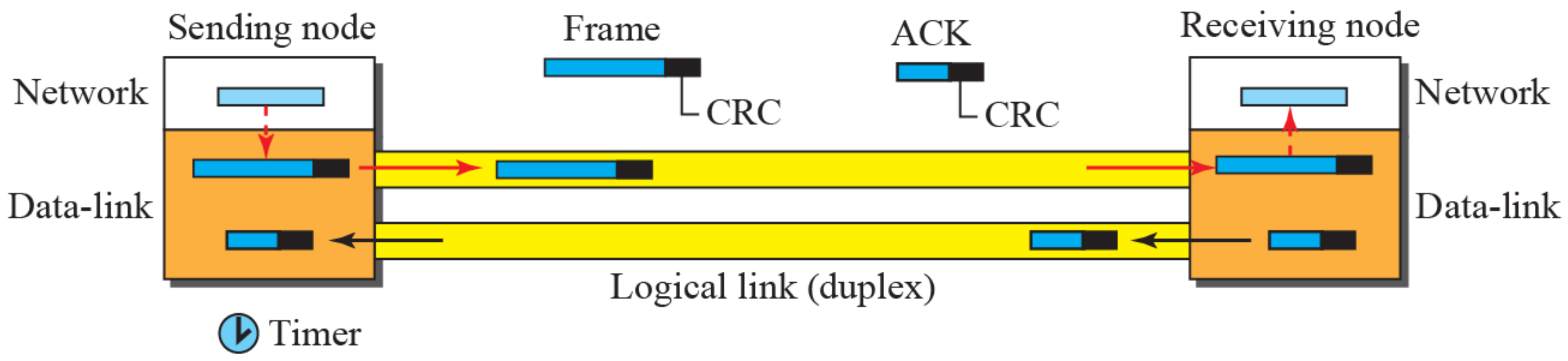
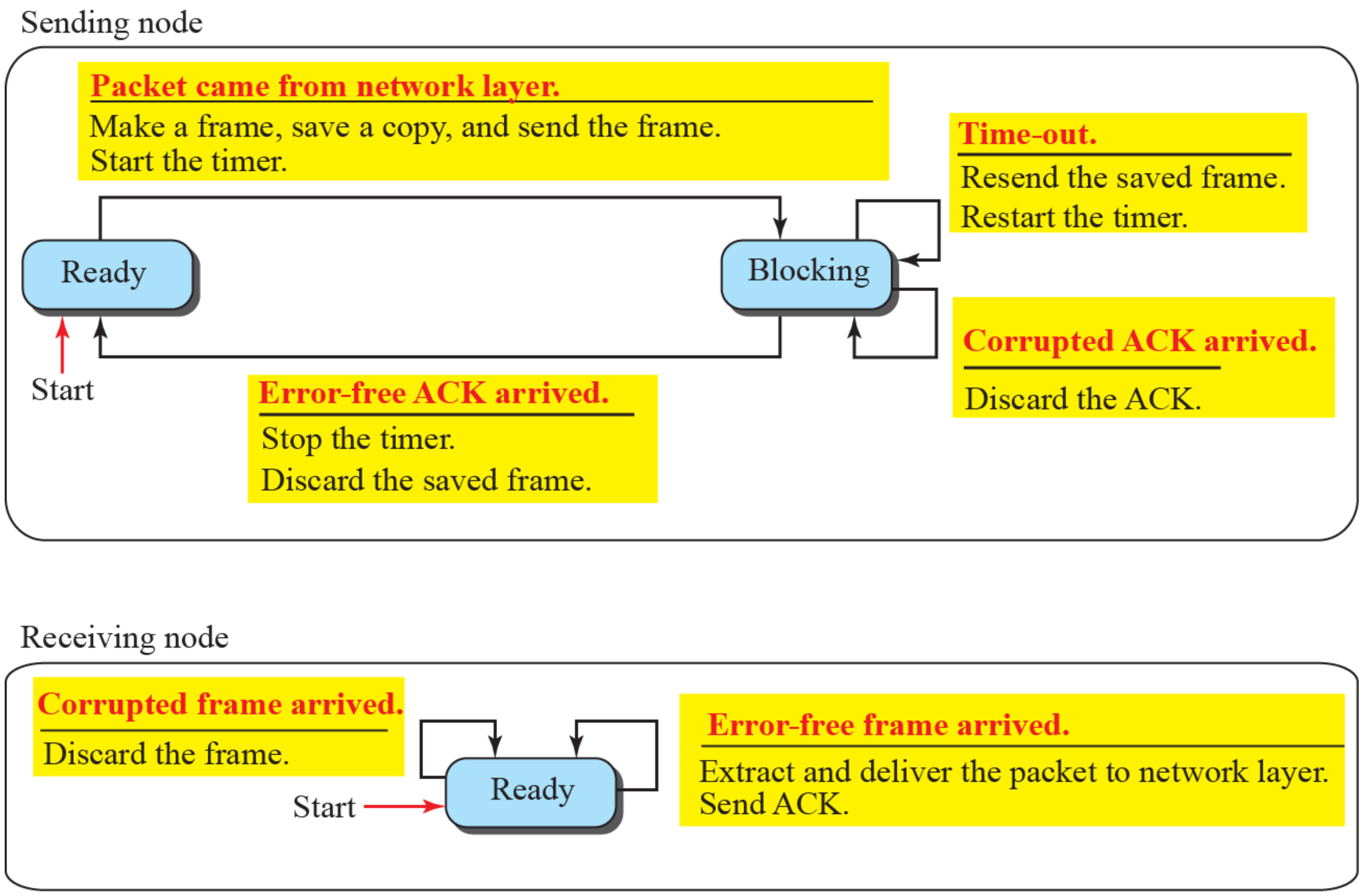


Figure 11.11: FSM for the stop-and-wait protocol



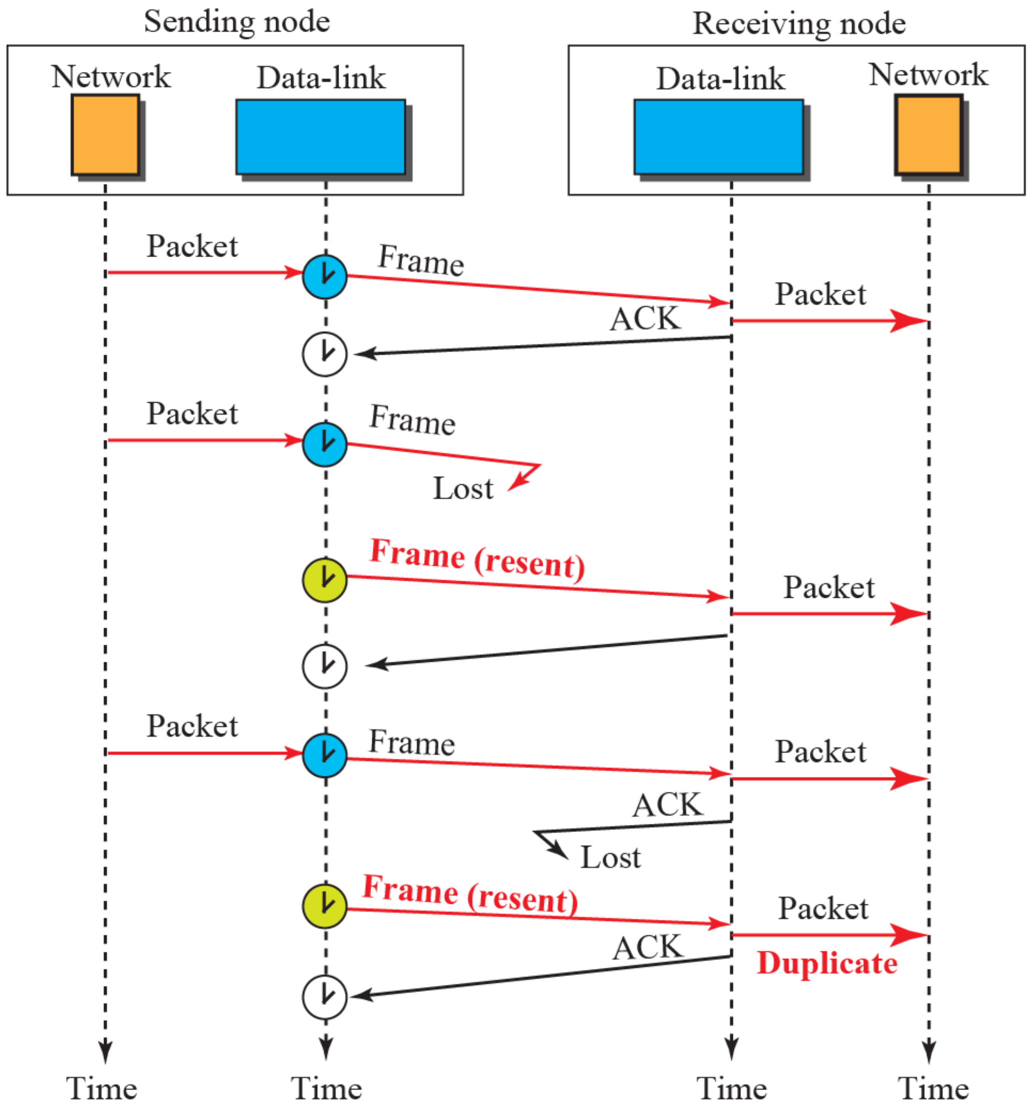
Example 11.3

Figure 11.12 shows an example. The first frame is sent and acknowledged. The second frame is sent, but lost. After time-out, it is resent. The third frame is sent and acknowledged, but the acknowledgment is lost. The frame is resent. However, there is a problem with this scheme. The network layer at the receiver site receives two copies of the third packet, which is not right. In the next section, we will see how we can correct this problem using sequence numbers and acknowledgment numbers.

Figure 11.12: Flow diagram for Example 111.3

Legend

- ● Start the timer.
- ○ Stop the timer.
- ● Restart a time-out timer.



Notes:
A lost frame means either lost or corrupted.
A lost ACK means either lost or corrupted.

Example 11.4

Figure 11.13 shows how adding sequence numbers and acknowledgment numbers can prevent duplicates. The first frame is sent and acknowledged. The second frame is sent, but lost. After time-out, it is resent. The third frame is sent and acknowledged, but the acknowledgment is lost. The frame is resent.

Figure 11.13: Flow diagram for Example 111.4

Legend

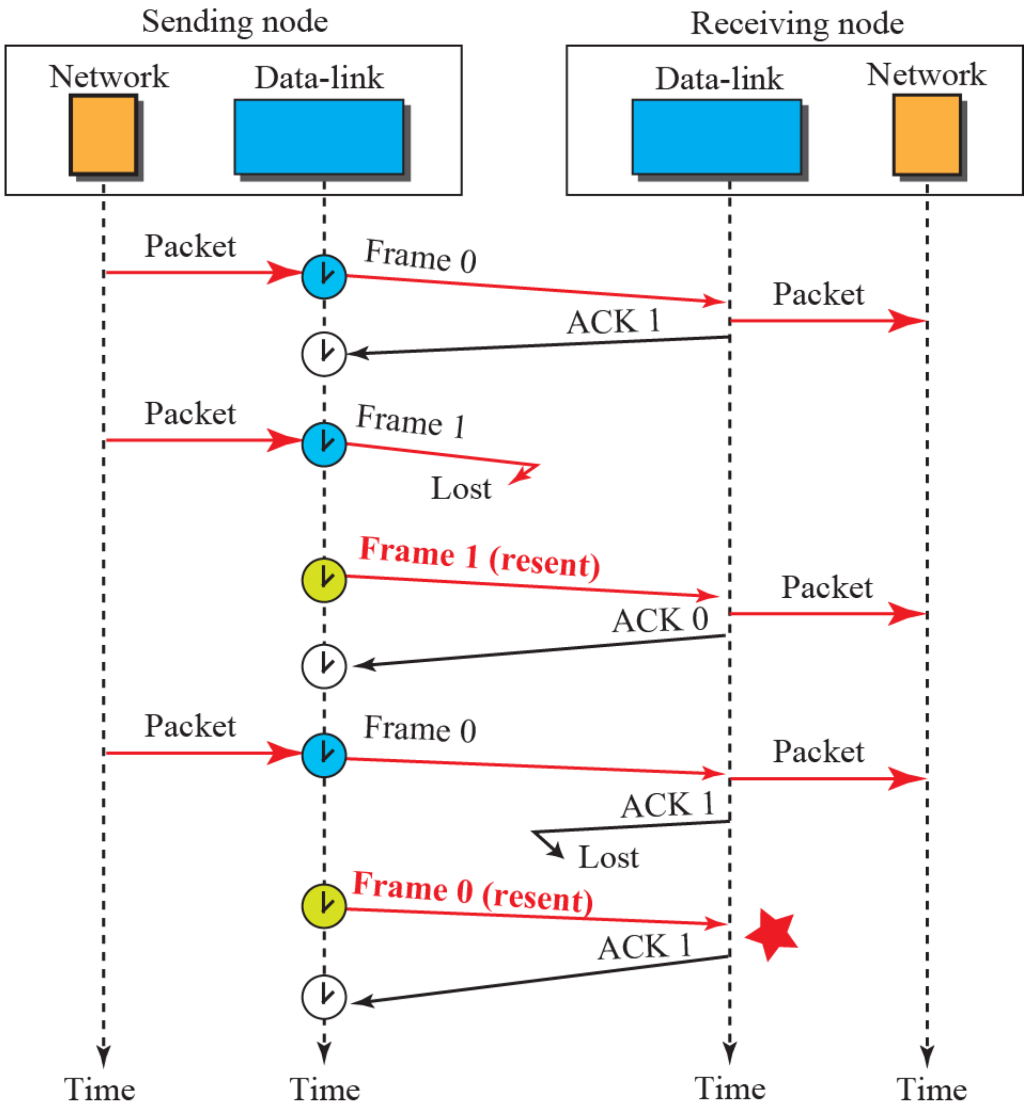
- ● Start the timer.
- ○ Stop the timer.
- ● Restart a time-out timer.

Notes:

A lost frame means either lost or corrupted.

A lost ACK means either lost or corrupted.

Frame 0 is discarded because the receiver expects frame 1.



11.2.3 Piggybacking

The two protocols we discussed in this section are designed for unidirectional communication, in which data is flowing only in one direction although the acknowledgment may travel in the other direction. Protocols have been designed in the past to allow data to flow in both directions. However, to make the communication more efficient, the data in one direction is piggybacked with the acknowledgment in the other direction.