Data Communications and Networking Fourth Edition

Forouzan

Chapter 11

Data Link Control

- ✓ The two main functions of the data link layer are data link control and media access control.
- ✓ Data link control, deals with the design and procedures for communication between two adjacent nodes.
  - ✓ Node-to-node communication.
  - ✓ Framing,
  - ✓ Flow and error control, and software implemented protocols that provide smooth and reliable transmission of frames between nodes.
- ✓ Media access control deals how to share the link between two or more nodes.

11-1 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 & a destination address. The destination address defines where the packet is to go; the sender address helps the recipient acknowledge the receipt.***

Fixed-Size Framing: In this there is no need for defining the boundaries of the frames, the size itself can be used as a delimiter.

Ex: ATM wide area network which uses frames of fixed size called cells.

Variable-Size Framing : In this, we need a way to define the end of the frame and the beginning of the next. Ex: LAN

Historically 2 approaches were used for variable size framing: **Character-oriented & bit-oriented**.

Figure 11.1 *A frame in a character-oriented protocol*

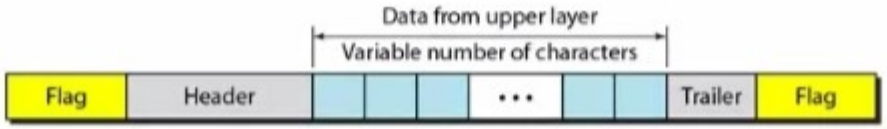
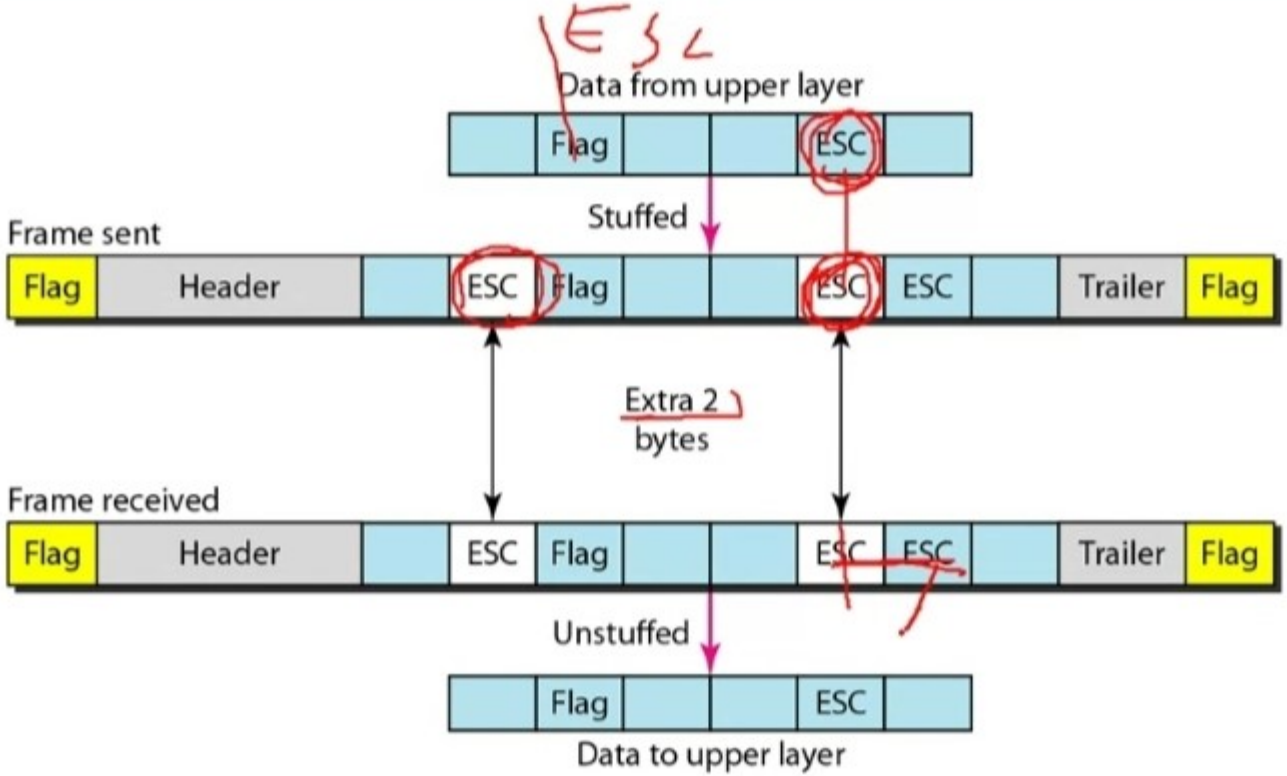


Figure 11.2 *Byte stuffing and unstuffing*



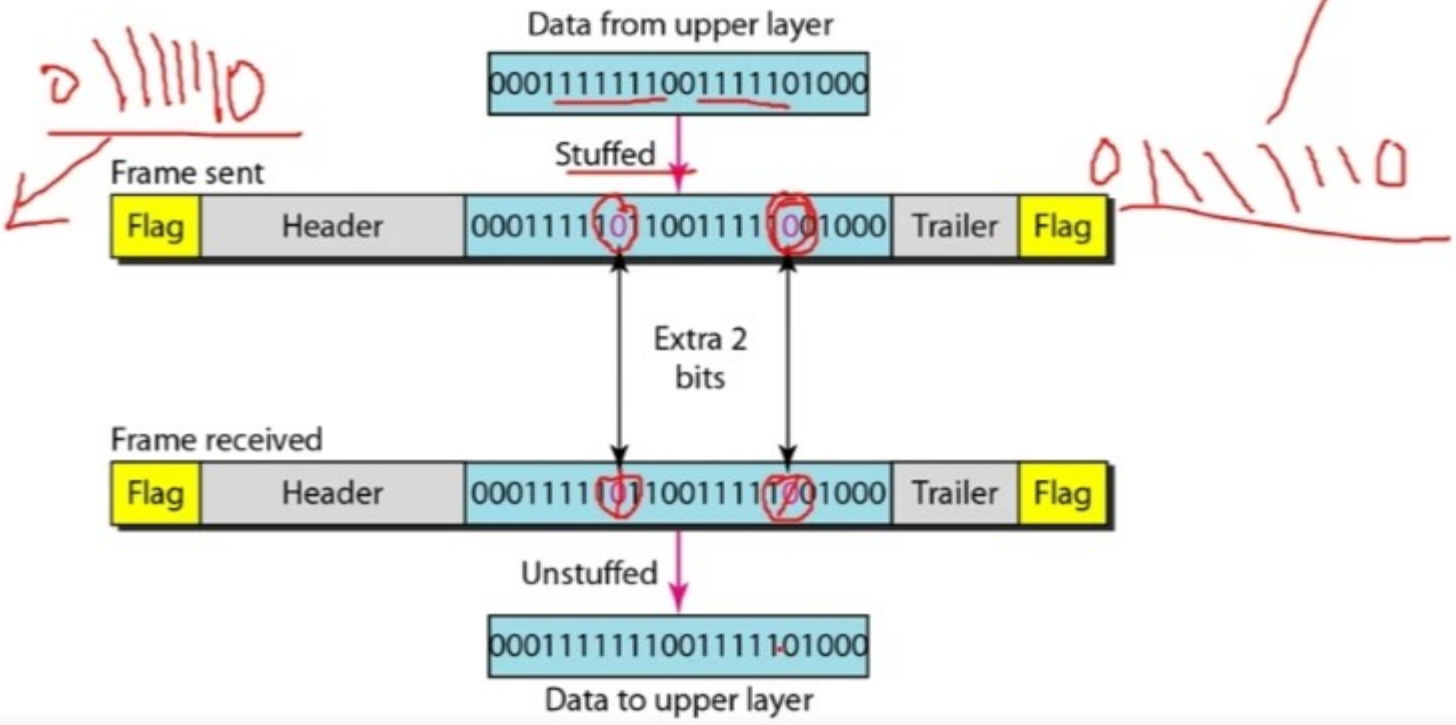
Note

Byte stuffing is the process of adding 1 extra byte whenever there is a flag or escape character in the text.

Note

Bit stuffing is the process of adding one extra 0 whenever five consecutive 1s follow a 0 in the data, so that the receiver does not mistake the pattern 0111110 for a flag.

Figure 11.4 *Bit stuffing and unstuffing*



11-2 FLOW AND ERROR CONTROL

The most important responsibilities of the data link layer are flow control and error control. Collectively, these functions are known as data link control.

Topics discussed in this section:

Flow Control
Error Control

11-2 FLOW AND ERROR CONTROL

Flow Control:

- ✓ Any receiving device has a limited speed at which it can process incoming data and a limited amount of memory in which to store incoming data.
- ✓ The flow of data must not be allowed to overwhelm the receiver.
- ✓ The receiver must be able to tell the sender to halt transmission until it is once again able to receive.
- ✓ Flow control refers to a set of procedures used to restrict the amount of data that the sender can send before waiting for acknowledgement.

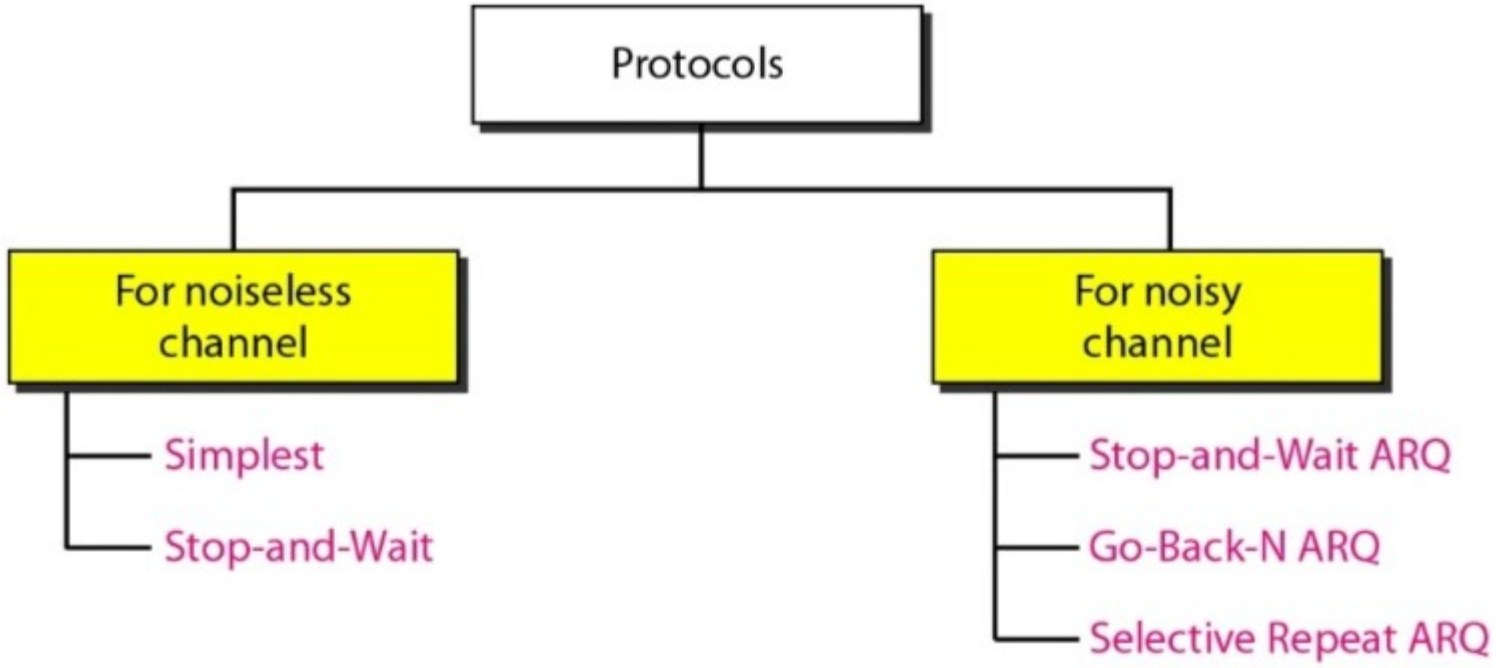
Error Control:

- ✓ Errors occur due to noises in the channel.
- ✓ Error control is both error detection & error correction.
- ✓ In the data link layer error control refers to methods of error detection and retransmission.
- ✓ To sender should add certain amount of redundant bits to the data, based on which the receiver will be able to detect errors.

11-3 PROTOCOLS

Now let us see how the data link layer can combine framing, flow control, and error control to achieve the delivery of data from one node to another. The protocols are normally implemented in software by using one of the common programming languages. To make our discussions language-free, we have written in pseudocode a version of each protocol that concentrates mostly on the procedure instead of delving into the details of language rules.

Figure 11.5 *Taxonomy of protocols discussed in this chapter*



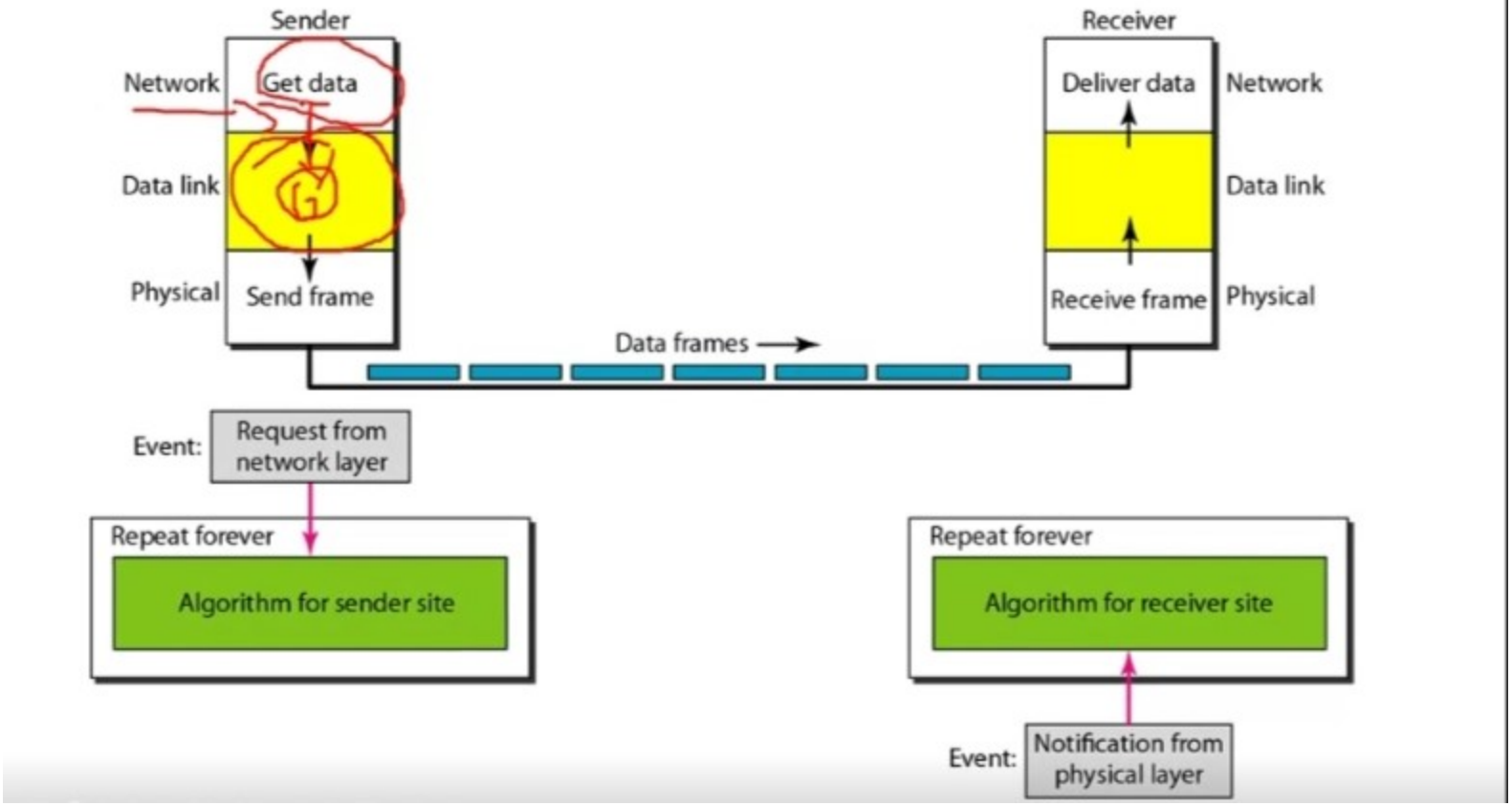
11-4 NOISELESS CHANNELS

Let us first assume we have an ideal channel in which no frames are lost, duplicated, or corrupted. We introduce two protocols for this type of channel.

Topics discussed in this section:

Simplest Protocol
Stop-and-Wait Protocol

Figure 11.6 *The design of the simplest protocol with no flow or error control*



Algorithm 11.1 *Sender-site algorithm for the simplest protocol*

```
1  while(true)                          // Repeat forever
2  {
3    WaitForEvent();                    // Sleep until an event occurs
4    if(Event(RequestToSend))           //There is a packet to send
5    {
6       GetData();
7       MakeFrame();
8       SendFrame();                    //Send the frame
9    }
10 }
```

Figure 11.7 *Flow diagram for Example 11.1*

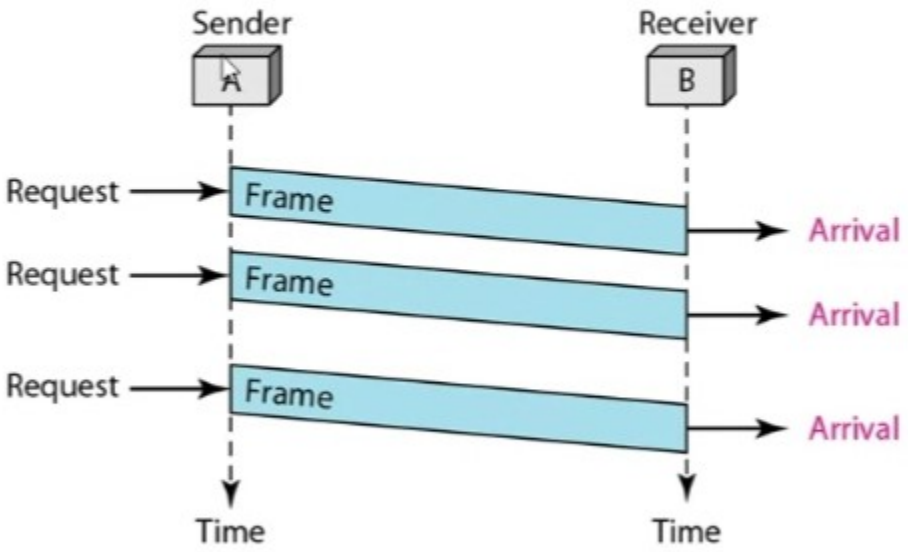


Figure 11.8 *Design of Stop-and-Wait Protocol*