OVERVIEW OF DATA LINK CONTROL

The data link control (DLC) deals with procedures for communication between two adjacent nodes.ie: node-to-node communication.No matter whether the link is dedicated or broadcast.

The functions of Data link control are as follows:

Framing, flow and error control.

Framing

Data transmission in the physical layer means moving bits in the form of a signal from the source to the destination.The physical layer provides bit synchronization to ensure that the sender and receiver use the same bit durations and timing.

The main function of Framing:

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.

Frames can be of two types: Fixed or variable size.

In fixed-size framing, there is no need for defining the boundaries of the frames; the size itself can be used as a delimiter. An example of this type of framing is the ATM WAN, which uses frames of fixed size called cells.

In variable-size framing, we need a way to define the end of one frame and the beginning of the next.

Character-Oriented Framing

In character-oriented (or byte-oriented) framing as shown in figure 2.1.1, data to be carried are 8-bit characters from a coding system such as ASCII. Here 8 bit data is used.The header, normally carries the source and destination addresses and other control information, and the trailer, which carries error detection redundant bits, and multiples of 8 bits. To separate one frame from the next, an 8-bit (1-byte) flag is added at the beginning and the end of a frame. The flag, composed of protocol-dependent special characters, signals the start or end of a frame.

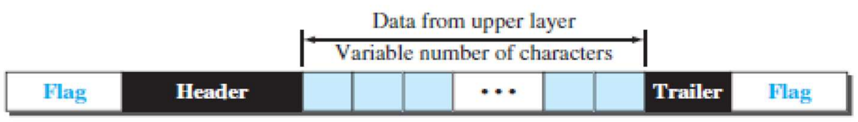

Fig2.1.1: Frame in character oriented protocol.

[Source : "Data Communications and Networking" by Behrouz A. Forouzan,Page-295]

We can also send other types of information such as graphs, audio, and video; any character used for the flag could also be part of the information. If this happens, the receiver, when it encounters this pattern in the middle of the data, thinks it has reached the end of the frame.

To overcome this problem, a byte-stuffing strategy was added to character oriented framing.

In byte stuffing (or character stuffing), a special byte is added to the data section of the frame when there is a character with the same pattern as the flag.

The data section is stuffed with an extra byte as in figure 2.1.2 . This byte is called the escape character (ESC) and has a predefined bit pattern. Whenever the receiver encounters the ESC character, it removes it from the data section and treats the next character as data, not as a delimiting flag.

.

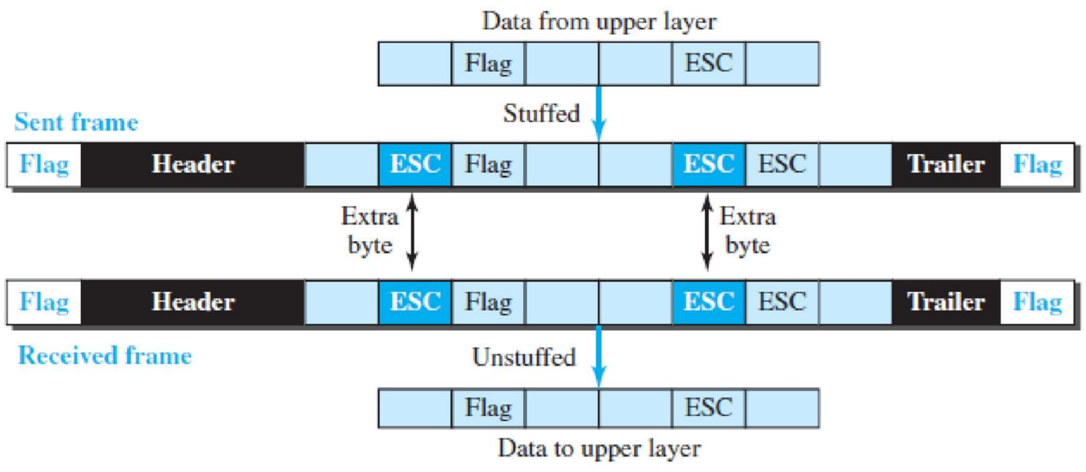

Fig2.1.2: Byte stuffing and unstuffing .

[Source : "Data Communications and Networking" by Behrouz A. Forouzan,Page-295]

Bit-Oriented Framing

In bit-oriented framing, the data section of a frame is a sequence of bits to be interpreted by the upper layer as text, graphic, audio, video, and so on.In addition to headers ,we need a delimiter to separate one frame from the other. Most protocols use a special 8-bit pattern flag, 01111110, as the delimiter to define the beginning and the end of the frame, as shown in Figure .

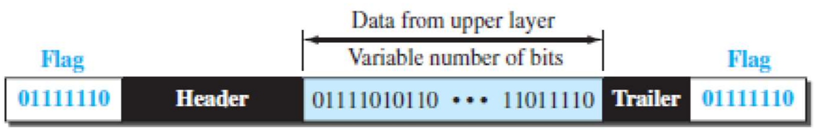

Fig2.1.3: Bit oriented framing.

[Source : "Data Communications and Networking" by Behrouz A. Forouzan,Page-296]

If the flag pattern appears in the data, we need to inform the receiver that this is not the end of the frame.By stuffing 1 single bit(instead of 1 byte) we can prevent the pattern from looking like a flag. This strategy is called bit stuffing.

In bit stuffing, if a 0 and five consecutive 1 bits are encountered, an extra0 is added as shown in figure 2.1.3. This extra stuffed bit is eventually removed from the data by the receiver.Bit stuffing is the process of adding one extra 0 whenever five consecutive 1s follow a 0in the data, so that the receiver does not mistake the pattern 0111110 for a flag.

Flow and Error Control

Flow Control

Whenever an entity produces items and another entity consumes them, there should be a balance between production and consumption rates. If the items are produced faster than they can be consumed, the consumer can be overwhelmed and may need to discard some items.

Flow control is implemented to prevent traffic.

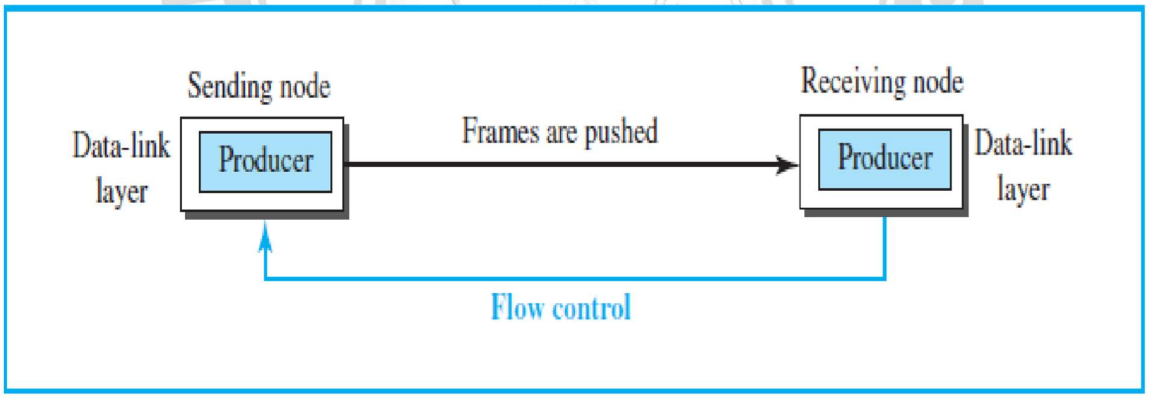

Fig2.1.4: Flow control at the data-link layer .
[Source : "Data Communications and Networking" by Behrouz A. Forouzan,Page-297]

In the figure2.1.4, the data-link layer at the sending node tries to push frames towards the data-link layer at the receiving node. If the receiving node cannot process and deliver the packet to its network at the same rate that the frames arrive, it becomes over whelmed (traffic) with frames.

Flow control in this case can be feedback from the receiving node to the sending node to stop or slow down pushing frames.

Buffers

Although flow control can be implemented in several ways, one of the solutions is normally to use two buffers; one at the sending data-link layer and the other at the receiving data-link layer.

A buffer is a set of memory locations that can hold packets at the sender and receiver.

Error Control

Error control at the data-link layer is very simple and implemented using one of the following two methods. In both methods, a CRC is added to the frame header by the sender and checked by the receiver.

In the first method, if the frame is corrupted, it is silently discarded; if it is not corrupted,the packet is delivered to the network layer. This method is used mostly in wired LANs such as Ethernet.

In the second method, if the frame is corrupted, it is silently discarded; if it is not corrupted, an acknowledgment is sent (for the purpose of both flow and error control)to the sender.

DATA-LINK LAYER PROTOCOLS

The behavior of a data-link-layer protocol is shown as a finite state machine (FSM). An FSM is a machine with a finite number of states. The machine is always in one of the states until an event occurs.

Each event has two reactions: defining the list (possibly empty) of actions to be performed and determining the next state (which can be the same as the current state). One of the states must be defined as the initial state, the state in which the machine starts when it turns on.

Here rounded-corner rectangles are used to show states, colored text to show events, and regular black text to show actions.A horizontal line is used to separate the event from the actions. The arrow shows the movement to the next state.There are only three possible events and three possible actions.

The machine starts in state I. If event 1 occurs, the machine performs actions 1 and 2 and moves to state II. When the machine is in state II, two events may occur.

If event 1 occurs, the machine performs action 3 and remains in the same state, state II. If event 3 occurs, the machine performs no action, but move to state I.

Simple Protocol

This is a simple protocol with neither flow nor error control. Assume that the receiver can immediately handle any frame it receives. In other words, the receiver can be free of congestion with incoming frames. Figure2.1.5 shows the layout for this protocol.

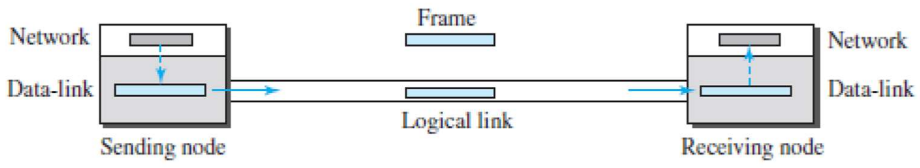

Fig2.1.5: Layout of simple protocol.
[Source : "Data Communications and Networking" by Behrouz A. Forouzan,Page-300]

The data-link layer at the sender gets a packet from its network layer, makes a frame out of it, and sends the frame. The data-link layer at the receiver receives a frame from the link, extracts the packet from the frame, and delivers the packet to its network layer.

FSMs (Description of Simple protocol function)

The sender should not send a frame until its network layer has a message to send.The receiver site cannot deliver a message to its network layer until a frame arrives. Requirements are based on two FSMs. Each FSM has only one state, the ready state as shown in figure 2.1.6.

The sending machine remains in the ready state until a request comes from the process in the network layer. When this event occurs, the sending machine encapsulates the message in a frame and sends it to the receiving machine.

The receiving machine remains in the ready state until a frame arrives from the sending machine. When this event occurs, the receiving machine decapsulates the message out of the frame and delivers it to the process at the network layer.

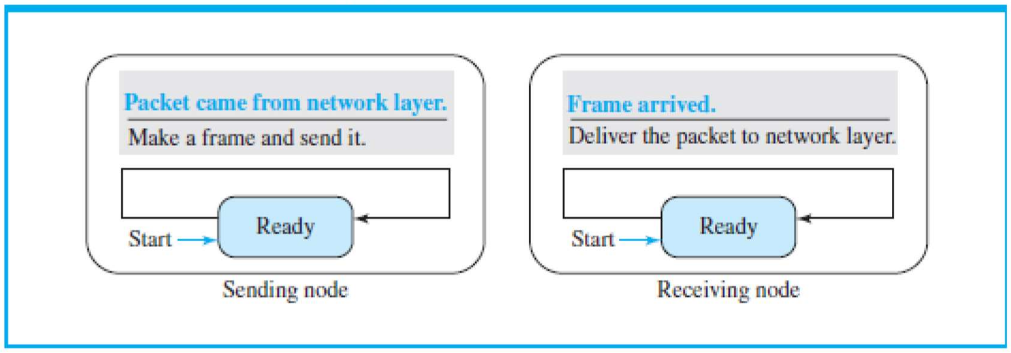
Packet came from network layer.
Make a frame and send it.
Start
Ready
Sending node
Frame arrived.
Deliver the packet to network layer.
Start
Ready
Receiving node

Fig2.1.6: FSM for simple protocol.
[Source : "Data Communications and Networking" by Behrouz A. Forouzan,Page-301]

Stop-and-Wait Protocol

Stop-and-Wait protocol, uses both flow and error control.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 shown in figure 2.1.7.

When a frame arrives at the receiver site, it is checked. If its CRC is incorrect, the frame is corrupted and silently discarded. The silence of the receiver is a signal for the sender that a frame was either corrupted or lost.

Every time the sender sends a frame, it starts a timer. If an acknowledgment arrives before the timer expires, the timer is stopped and the sender sends the next frame (if it has one to send). If the timer expires, the sender resends the previous frame, assuming that the frame was either lost or corrupted. This means that the sender needs to keepa copy of the frame until its acknowledgment arrives.

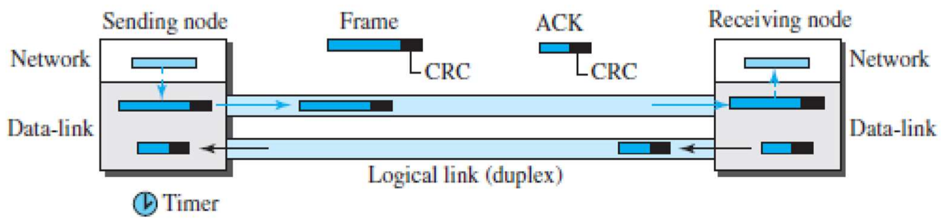

Fig2.1.7: Stop and wait protocol .

[Source : "Data Communications and Networking" by Behrouz A. Forouzan,Page-302]

FSM (Finite State Machine)

Figure 2.1.8 shows the FSMs for the Stop-and-Wait protocol. Sender and receiver states are described as follows.

Sender States

The sender is initially in the ready state, but it can move between the ready and blocking state.

Ready State.When the sender is in this state, it is only waiting for a packet from the network layer. If a packet comes from the network layer, the sender creates a frame, saves a copy of the frame, starts the only timer and sends the frame. The sender then moves to the blocking state.

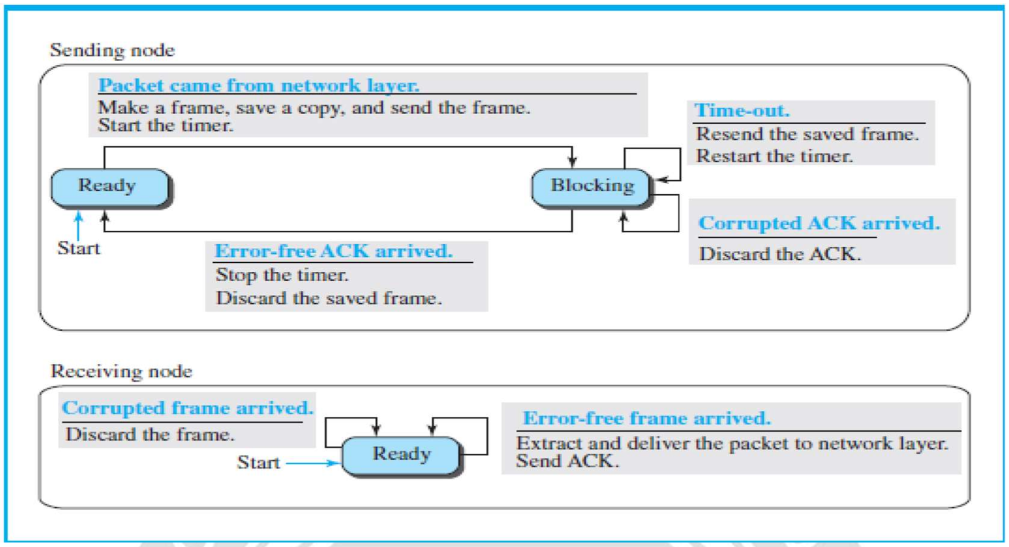

Fig2.1.8: FSM for the stop and wait protocol.

[Source : "Data Communications and Networking" by Behrouz A. Forouzan,Page-302]

Blocking State .1.When the sender is in this state, three events can occur:

If a time-out occurs, the sender resends the saved copy of the frame and restarts the timer.

2.If a corrupted ACK arrives, it is discarded.

3.If an error-free ACK arrives, the sender stops the timer and discards the saved copy of the frame. It then moves to the ready state.

Receiver

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

1.If an error-free frame arrives, the message in the frame is delivered to the network layer and an ACK is sent. If a corrupted frame arrives, the frame is discarded.