3.5.5 Compounding High- and Low-Resolution Packets

- Another solution is to create a duplicate of each packet with a low-resolution redundancy and combine the redundant version with the next packet.
- For example (Figure 10.22):

  We can create 4 low-resolution packets out of 5 high-resolution packets and send them (Fig 10.22).
- If a packet is lost, we can use the low-resolution version from the next packet.
- In this method, if the last packet is lost, it cannot be recovered, but we use the low-resolution version of a packet if the lost packet is not the last one.
- The audio and video reproduction does not have the same quality, but the lack of quality is not recognized most of the time.

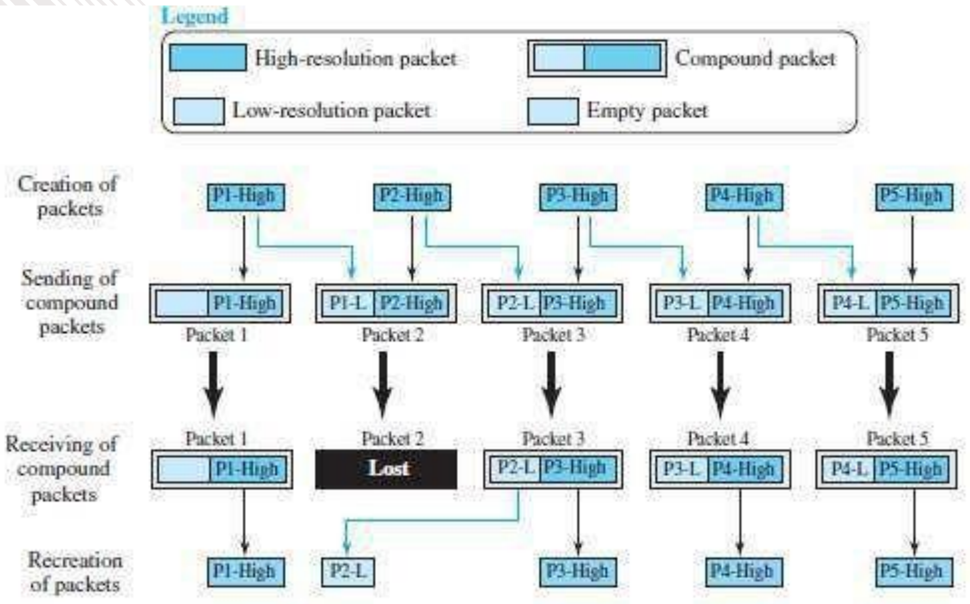

Figure 10.22 *Compounding high- and low-resolution packets*

DATA LINK CONTROL

3.6 DLC SERVICES

- The data link control (DLC) deals with procedures for communication between two adjacent nodes i.e. node-to-node communication.
- Data link control functions include 1) Framing and 2) Flow control and 3) Error control.

3.6.1 Framing

- A frame is a group of bits.
- Framing means organizing the bits into a frame that are carried by the physical layer.
- The data-link-layer needs to form frames, so that each frame is distinguishable from another.
- Framing separates a message from other messages by adding sender-address & destination-address.
- The destination-address defines where the packet is to go.

  The sender-address helps the recipient acknowledge the receipt.
- Q: Why the whole message is not packed in one frame?

Ans: Large frame makes flow and error-control very inefficient.

Even a single-bit error requires the re-transmission of the whole message.

When a message is divided into smaller frames, a single-bit error affects only that small frame. (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. In addition, each envelope defines the sender and receiver addresses since the postal system is a many-to-many carrier facility).

3.6.1.1 Frame Size • Two types of frames: **1) Fixed Size Framing**

 There is no need for defining boundaries of frames; the size itself can be used as a delimiter. For example: ATM WAN uses frames of fixed size called cells.

2) Variable Size Framing

 We need to define the end of the frame and the beginning of the next frame. Two approaches are used: 1) Character-oriented approach 2) Bit-oriented approach.

3.6.1.2 Character Oriented Framing

- Data to be carried are 8-bit characters from a coding system such as ASCII (Figure 11.1).
- The header and the trailer are also multiples of 8 bits.

  1) Header carries the source and destination-addresses and other control information. 2) Trailer carries error-detection or error-correction redundant bits.
- To separate one frame from the next frame, an 8- bit (I-byte) flag is added at the beginning and the end of a frame.
- The flag is composed of protocol-dependent special characters.
- The flag signals the start or end of a frame.

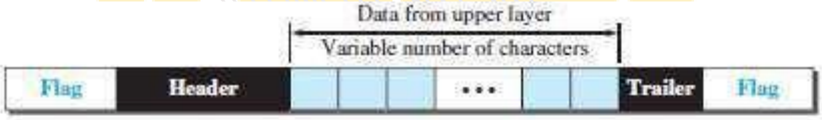

Figure 11.1 *A frame in a character-oriented protocol*

- Problem:
  - Character-oriented framing is suitable when only text is exchanged by the data-link-layers. However, if we send other type of information (say audio/video), then any pattern used for the flag can also be part of the information.
  - If the flag-pattern appears in the data-section, the receiver might think that it has reached the end of the frame.

Solution: A byte-stuffing is used.

(Byte stuffing character stuffing)

- In byte 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. This byte is called the escape character (ESC), which has a predefined bit pattern.
- When a receiver encounters the ESC character, the receiver → removes ESC character from the data-section and → treats the next character as data, not a delimiting flag.
- Problem:
  - What happens if the text contains one or more escape characters followed by a flag?

- The receiver removes the escape character, but keeps the flag, which is incorrectly interpreted as the end of the frame. Solution:
- Escape characters part of the text must also be marked by another escape character (Fig 11.2).

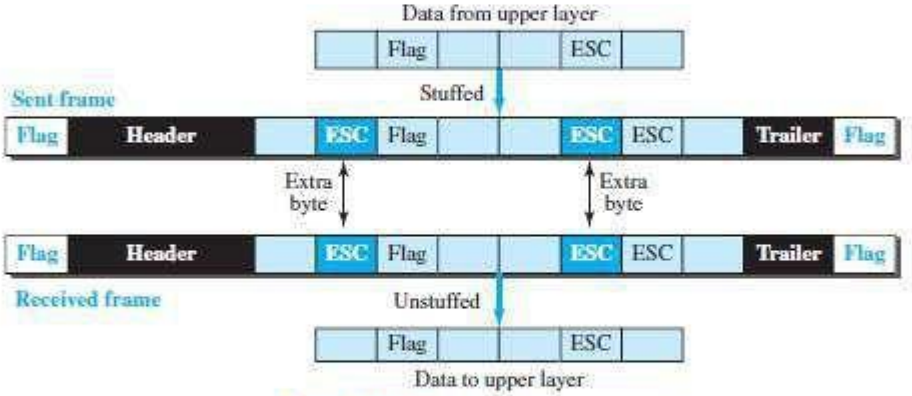

Figure 11.2 *Byte stuffing and unstuffing*

- In short, byte stuffing is the process of adding one extra byte whenever there is a flag or escape character in the text.

3.6.1.3 Bit Oriented Framing

- The data-section of a frame is a sequence of bits to be interpreted by the upper layer as text, audio, video, and so on.
- However, in addition to headers and trailers, we need a delimiter to separate one frame from theother.
- Most protocols use a special 8-bit pattern flag 01111110 as the delimiter to define the beginning and the end of the frame (Figure 11.3).

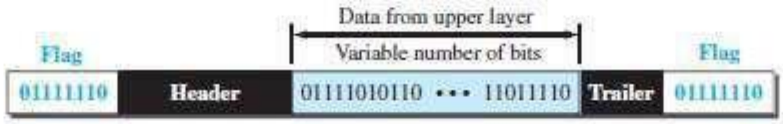

Figure 11.3 *A frame in a bit-oriented protocol*

- Problem:
  - If the flag-pattern appears in the data-section, the receiver might think that it has reached the end of the frame.

Solution: A bit-stuffing is used.

- In bit stuffing, if a 0 and five consecutive 1 bits are encountered, an extra 0 is added. This extra stuffed bit is eventually removed from the data by the receiver. (Figure 11.4).
- This guarantees that the flag field sequence does not inadvertently appear in the frame.

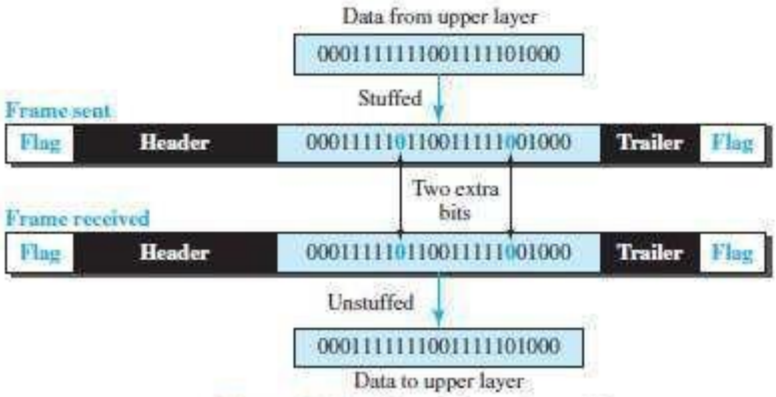

Figure 11.4 *Bit stuffing and unstuffing*

- In short, 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.

3.6.2 Flow Control and Error Control

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

3.6.2.1 Flow Control

- Whenever an entity produces items and another entity consumes them, there should be a balancebetween 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.
- We need to prevent losing the data items at the consumer site.

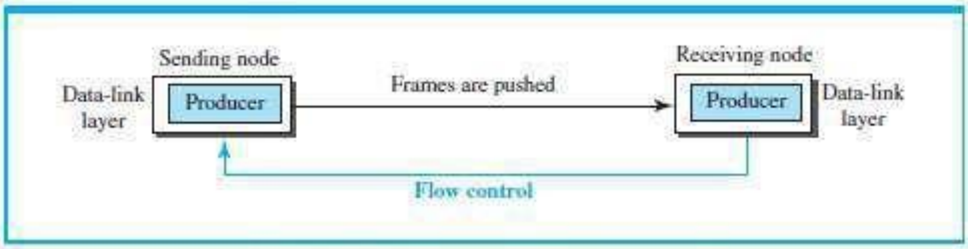

Figure 11.5 *Flow control at the data-link layer*

- At the sending node, the data-link layer tries to push frames toward the data-link layer at the receiving node (Figure 11.5).
- If the receiving node cannot process and deliver the packet to its network at the same rate that theframes arrive, it becomes overwhelmed with frames.
- Here, flow control can be feedback from the receiving node to the sending node to stop or slow down pushing frames.

3.6.2.1.1 Buffers

- Flow control can be implemented by using buffer.
- A buffer is a set of memory locations that can hold packets at the sender and receiver.
- Normally, two buffers can be used. 1) First buffer at the sender.
  2) Second buffer at the receiver.
- The flow control communication can occur by sending signals from the consumer to the producer.
- When the buffer of the receiver is full, it informs the sender to stop pushing frames.

3.6.2.2 Error Control

- Error-control includes both error-detection and error-correction.
- Error-control allows the receiver to inform the sender of any frames lost/damaged in transmission.
- A CRC is
  → added to the frame header by the sender and → checked
  by the receiver.
- At the data-link layer, error control is normally implemented using one of the following two methods.
  1) First method: If the frame is corrupted, it is discarded;
  If the frame is not corrupted, the packet is delivered to the network layer.
  This method is used mostly in wired LANs such as Ethernet.

2) Second method: If the frame is corrupted, it is discarded;

If the frame is not corrupted, an acknowledgment is sent to the sender.

Acknowledgment is used for the purpose of both flow and error control.

3.6.2.2.1 Combination of Flow and Error Control

- Flow and error control can be combined.
- The acknowledgment that is sent for flow control can also be used for error control to tell the sender the packet has arrived uncorrupted.
- The lack of acknowledgment means that there is a problem in the sent frame.
- A frame that carries an acknowledgment is normally called an ACK to distinguish it from the dataframe.

3.6.3 Connectionless and Connection-Oriented

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

1) Connectionless Protocol

- Frames are sent from one node to the next without any relationship between the frames; each frame is independent.
- The term connectionless does not mean that there is no physical connection (transmission medium) between the nodes; it means that there is no connection between frames.
- The frames are not numbered and there is no sense of ordering.
- Most of the data-link protocols for LANs are connectionless protocols.

2) Connection Oriented Protocol

- A logical connection should first be established between the two nodes (setup phase).
- After all frames that are somehow related to each other are transmitted (transfer phase), the logical connection is terminated (teardown phase).
- The frames are numbered and sent in order.
- If the frames are not received in order, the receiver needs to wait until all frames belonging to the same set are received and then deliver them in order to the network layer.
- Connection oriented protocols are rare in wired LANs, but we can see them in some point-to- point protocols, some wireless LANs, and some WANs.

3.7 DATA LINK LAYER PROTOCOLS

- Traditionally 2 protocols have been defined for the data-link layer to deal with flow and error control: 1) Simple Protocol and 2) Stop-and-Wait Protocol.
- The behavior of a data-link-layer protocol can be better shown as a finite state machine (FSM).
- An FSM is a machine with a finite number of states (Figure 11.6).
- The machine is always in one of the states until an event occurs.
- Each event is associated with 2 reactions:
    1) Defining the list (possibly empty) of actions to be performed.
    2) 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.

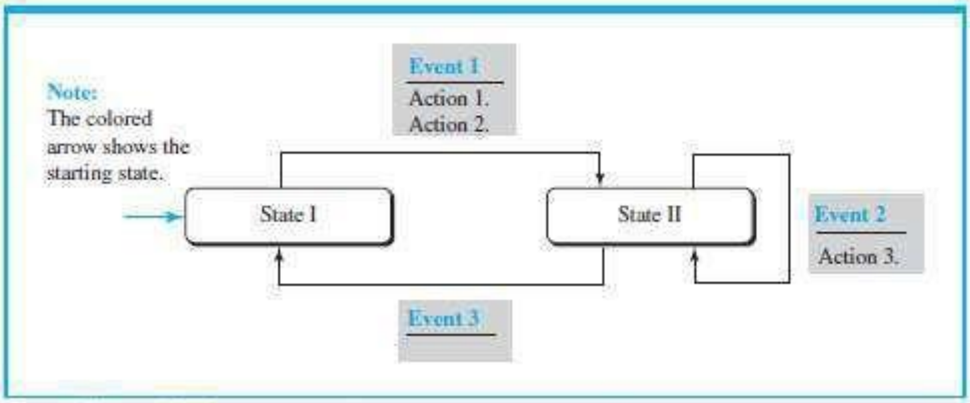

Figure 11.6 *Connectionless and connection-oriented service represented as FSMs*

3.7.1 Simplest Protocol

- Assumptions:
    - The protocol has no flow-control or error-control.
    - The protocol is a unidirectional protocol (in which frames are traveling in only one direction). The receiver can immediately handle any frame it receives.

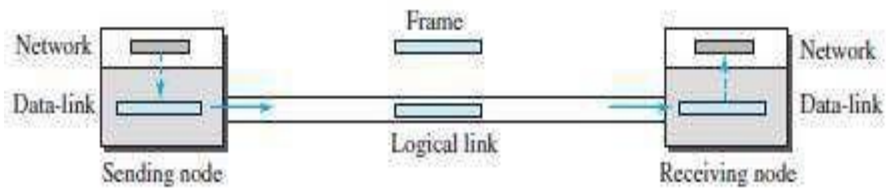

Figure 11.7 *Simple protocol*

3.7.1.1 Design

- Here is how it works (Figure 11.7):

1) At Sender

¤ The data-link-layer

→ gets data from its network-layer → makes a frame out of the data and → sends the frame.

2) At Receiver

¤ The data-link-layer

→ receives a frame from its physical layer →

extracts data from the frame and

→ delivers the data to its network-layer.

 Data-link-layers of sender & receiver provide transmission services for their network-layers. Data-link-layers use the services provided by their physical layers for the physical transmission of bits.

3.7.1.2 FSMs

- Two main requirements:
  1) The sender-site cannot send a frame until its network-layer has a data packet to send.
  2) The receiver-site cannot deliver a data packet to its network-layer until a frame arrives.
- These 2 requirements are shown using two FSMs.
- Each FSM has only one state, the ready state.

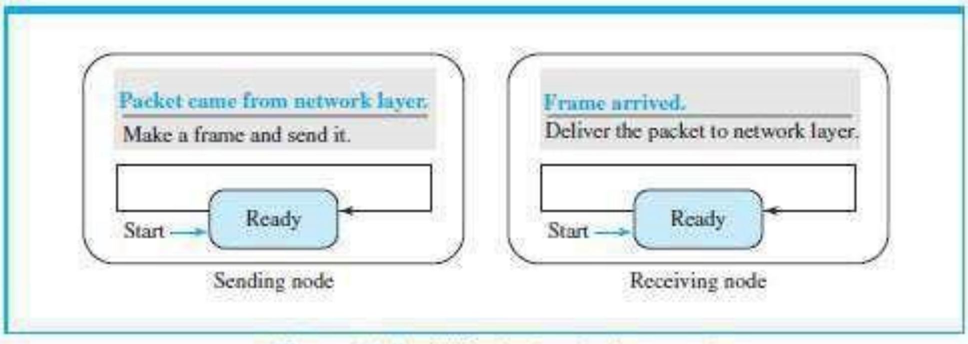

Figure 11.8 *FSMs for the simple protocol*

- Here is how it works (Figure 11.8):

1) At Sending Machine

 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.

2) At 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.

Example 3.6

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.

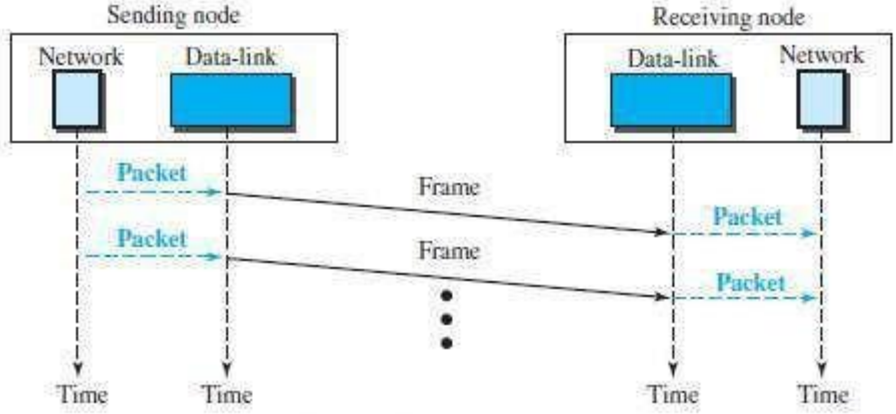

Figure 11.9 *Flow diagram*

3.7.2 Stop & Wait Protocol

- This uses both flow and error control.
- Normally, the receiver has limited storage-space.
- If the receiver is receiving data from many sources, the receiver may

  → be overloaded with frames &

  → discard the frames.
- To prevent the receiver from being overloaded with frames, we need to tell the sender to slow down.

3.7.2.1 Design

1) At Sender

- The sender

  → sends one frame & starts a timer

  → keeps a copy of the sent-frame and

  → waits for ACK-frame from the receiver (okay to go ahead).
- Then,

  1) If an ACK-frame arrives before the timer expires, the timer is stopped and the sender sends the next frame.

     Also, the sender discards the copy of the previous frame.

  2) If the timer expires before ACK-frame arrives, the sender resends the previous frame and restarts the timer

2) At Receiver

- To detect corrupted frames, a CRC is added to each data frame. When a frame arrives at the receiver-site, the frame is checked.
- If frame's CRC is incorrect, the frame is corrupted and discarded.
- The silence of the receiver is a signal for the sender that a frame was either corrupted or lost.

3.7.2.2 FSMs

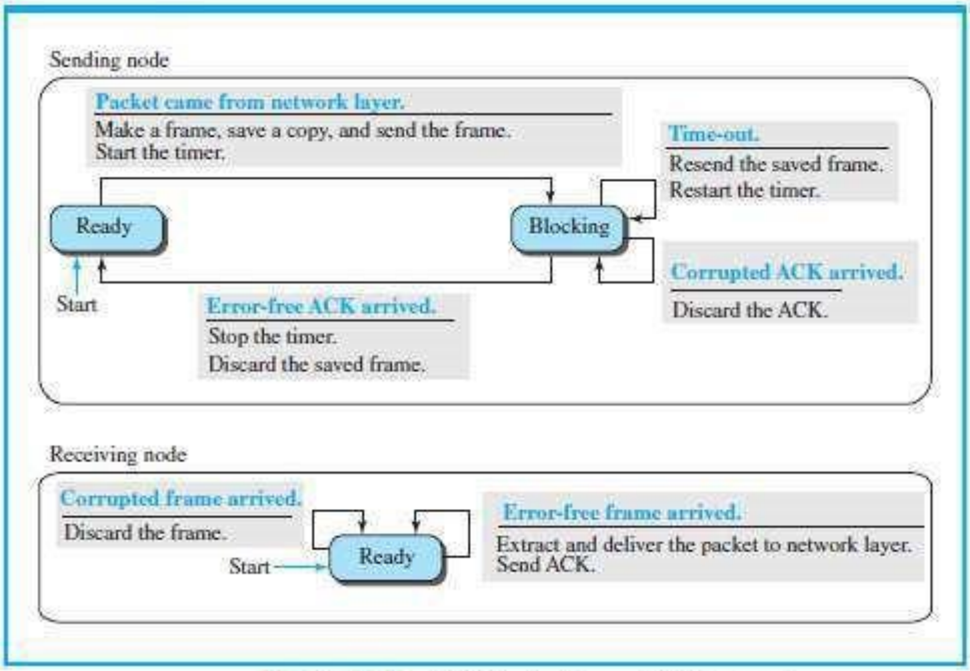

Figure 11.11 *FSM for the Stop-and-Wait protocol*

- Here is how it works (Figure 11.11):

1) Sender States

- Sender is initially in the ready state, but it can move between the ready and blocking state.

i) 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.

ii) Blocking State: When the sender is in this state, three events can occur:

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

b) If a corrupted ACK arrives, it is discarded.

c) 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.

2) Receiver

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

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

b) If a corrupted frame arrives, the frame is discarded.

Example 3.7

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.

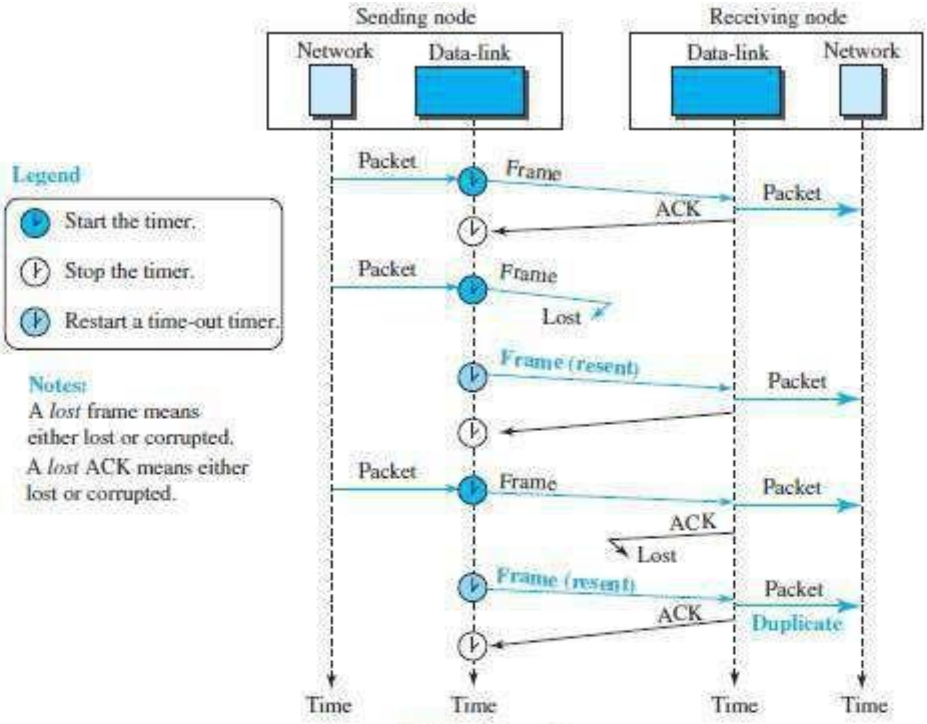

Figure 11.12 *Flow diagram*

3.7.2.3 Sequence and Acknowledgment Numbers • Q:
How to deal with corrupted-frame?

Ans: If the corrupted-frame arrives at the receiver-site, then the frame is simply discarded.

• Q: How to deal with lost-frames?

Ans: If the receiver receives out-of-order data-frame, then it means that frames were lost. .'. The lost-frames need to be resent.

• Problem in Stop and Wait protocols:

1) There is no way to identify a frame.

2) The received-frame could be the correct one, or a duplicate, or a frame out of order.

Solution: 1) Use sequence-number for each data frame.

2) Use Acknowledgment-number for each ACK frame.

Sequence Numbers

 Frames need to be numbered. This is done by using sequence-numbers.
A sequence-number field is added to the data-frame.

Acknowledgment Numbers

 An acknowledgment-number field is added to the ACK-frame. Sequence numbers are 0, 1, 0, 1, 0, 1, . . .

The acknowledgment numbers can also be 1, 0, 1, 0, 1, 0, …

- The acknowledgment-numbers always announce the sequence-number of the next frame expected by the receiver.
- For example,

  If frame-0 has arrived safely, the receiver sends an ACK-frame with acknowledgment-1 (meaning frame-1 is expected next).

Example 3.8

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.

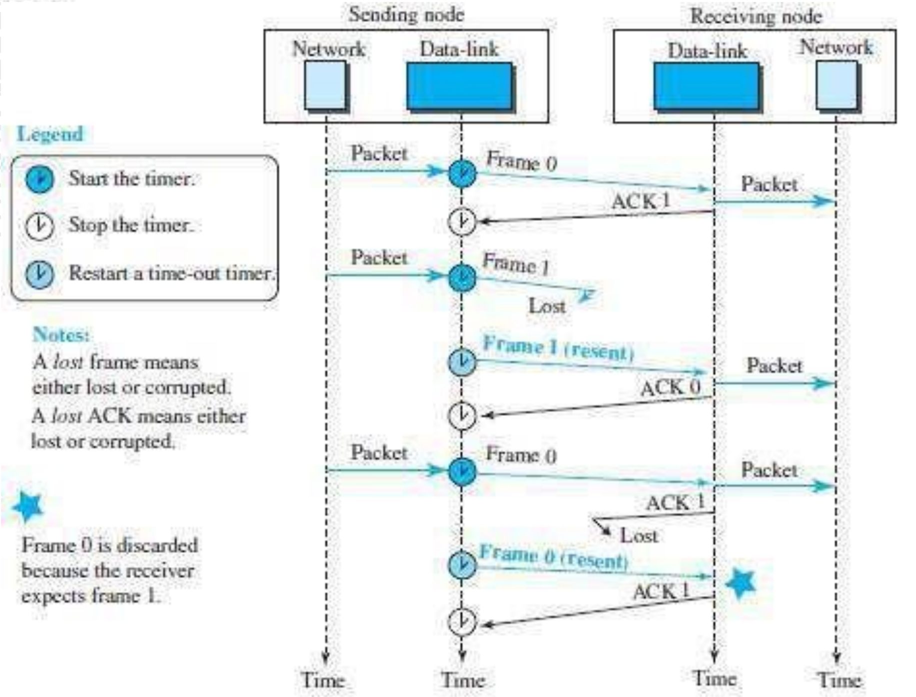

Figure 11.13 *Flow diagram*

3.7.3 Piggybacking

- A technique called piggybacking is used to improve the efficiency of the bidirectional protocols.
- The data in one direction is piggybacked with the acknowledgment in the other direction.
- In other words, when node A is sending data to node B, Node A also acknowledges the data received from node B.

3.8 High-Level Data Link Control (HDLC)

- HDLC is a bit-oriented protocol for communication over point-to-point and multipoint links.
- HDLC implements the ARQ mechanisms.

3.8.1 Configurations and Transfer Modes

- HDLC provides 2 common transfer modes that can be used in different configurations: 1) Normal response mode (NRM)

  2) Asynchronous balanced mode (ABM). **NRM**
- The station configuration is unbalanced (Figure 11.14).
- We have one primary station and multiple secondary stations.
- A primary station can send commands, a secondary station can only respond.
- The NRM is used for both point-to-point and multiple-point links.

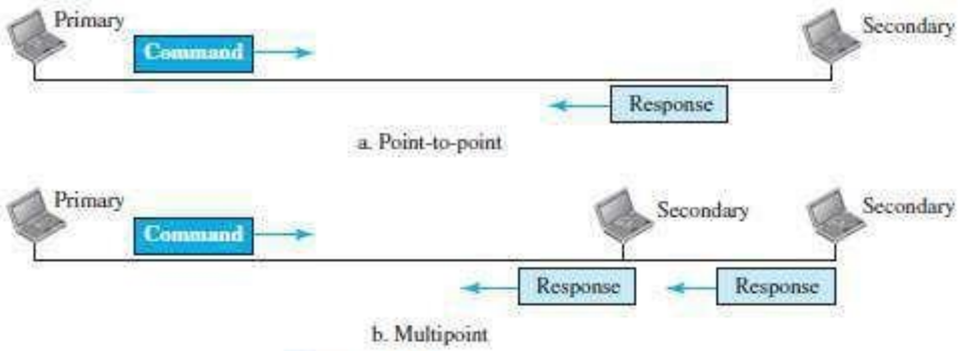

Figure 11.14 *Normal response mode*

ABM

- The configuration is balanced (Figure 11.15).
- Link is point-to-point, and each station can function as a primary and a secondary (acting as peers).
- This is the common mode today.

Figure 11.15 *Asynchronous balanced mode*

3.8.2 Framing

- To provide the flexibility necessary to support all the options possible in the modes andconfigurations, HDLC defines three types of frames:
  1) Information frames (I-frames): are used to transport user data and control information relating to user data (piggybacking).
  2) Supervisory frames (S-frames): are used only to transport control information.
  3) Unnumbered frames (U-frames): are reserved for system management.
     Information carried by U-frames is intended for managing the link itself.
- Each type of frame serves as an envelope for the transmission of a different type of message.

3.8.2.1 Frame Format

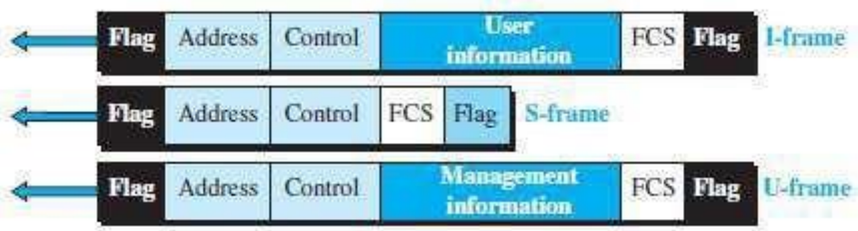

Figure 11.16 *HDLC frames*

• Various fields of HDLC frame are:

1) Flag Field

- This field has a synchronization pattern 01111110.
- This field identifies both the beginning and the end of a frame.

2) Address Field

- This field contains the address of the secondary station.
- If a primary station created the frame, it contains a to-address.
- If a secondary creates the frame, it contains a from-address.

 This field can be 1 byte or several bytes long, depending on the needs of the network.

3) Control Field

 This field is one or two bytes used for flow and error control.

4) Information Field

 This field contains the user's data from the network-layer or management information. Its length can vary from one network to another.

5) FCS Field

 This field is the error-detection field. (FCS Frame Check Sequence) This field can contain either a 2- or 4-byte standard CRC.

3.8.2.1.1 Control Fields of HDLC Frames

• The control field determines the type of frame and defines its functionality (Figure 11.17).

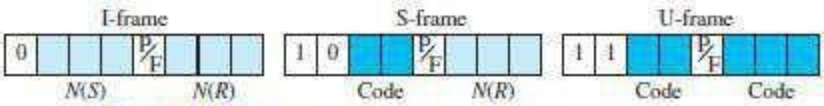

Figure 11.17 Control field format for the different frame types

1) Control Field for I-Frames

- I-frames are designed to carry user data from the network-layer.
- In addition, they can include flow and error-control information (piggybacking).
- The subfields in the control field are:
  1) The first bit defines the type.
     If the first bit of the control field is 0, this means the frame is an I-frame.
  2) The next 3 bits N(S) define the sequence-number of the frame.
     With 3 bits, we can define a sequence-number between 0 and 7
  3) The last 3 bits N(R) correspond to the acknowledgment-number when piggybacking is used.
  4) The single bit between N(S) and N(R) is called the P/F bit.
     The P/F field is a single bit with a dual purpose. It can mean poll or final.
     i) It means poll when the frame is sent by a primary station to a secondary (when theaddress field contains the address of the receiver).
     ii) It means final when the frame is sent by a secondary to a primary (when the address field contains the address of the sender).

2) Control Field for S-Frames

- Supervisory frames are used for flow and error-control whenever piggybacking is either impossible or inappropriate (e.g., when the station either has no data of its own to send or needs to send a command or response other than an acknowledgment).
- S-frames do not have information fields.
- The subfields in the control field are:
  1) If the first 2 bits of the control field is 10, this means the frame is an S-frame.
  2) The last 3 bits N(R) corresponds to the acknowledgment-number (ACK) or negative acknowledgment-number (NAK).
  3) The 2 bits called code is used to define the type of S-frame itself.
     With 2 bits, we can have four types of S-frames:

1) Receive Ready (RR) = 00

¤ This acknowledges the receipt of frame or group of frames. ¤ The value of N(R) is the acknowledgment-number. **2) Receive Not Ready (RNR) = 10**

¤ This is an RR frame with 1 additional function:

i) It announces that the receiver is busy and cannot receive more frames. ¤ It acts as congestion control mechanism by asking the sender to slow down. ¤ The value of N(R) is the acknowledgment-number.

3) ReJect (REJ) = 01

¤ It is a NAK frame used in Go-Back-N ARQ to improve the efficiency of the process. ¤ It informs the sender, before the sender time expires, that the last frame is lost or damaged.

¤ The value of N(R) is the negative acknowledgment-number.

4) Selective REJect (SREJ) = 11

¤ This is a NAK frame used in Selective Repeat ARQ.

¤ The value of N(R) is the negative acknowledgment-number.

3) Control Field for U-Frames

- Unnumbered frames are used to exchange session management and control information betweenconnected devices.
- U-frames contain an information field used for system management information, but not user data.
- Much of the information carried by U-frames is contained in codes included in the control field.
- U-frame codes are divided into 2 sections:

  i) A 2-bit prefix before the P/F bit ii) A 3-bit suffix after the P/F bit.
- Together, these two segments (5 bits) can be used to create up to 32 different types of U-frames.

Example 3.9

Figure 11.18 shows how U-frames can be used for connection establishment and connection release. Node A asks for a connection with a set asynchronous balanced mode (SABM) frame; node B gives a positive response with an unnumbered acknowledgment (UA) frame. After these two exchanges, data can be transferred between the two nodes (not shown in the figure). After data transfer, node A sends a DISC (disconnect) frame to release the connection; it is confirmed by node B responding with a UA (unnumbered acknowledgment).

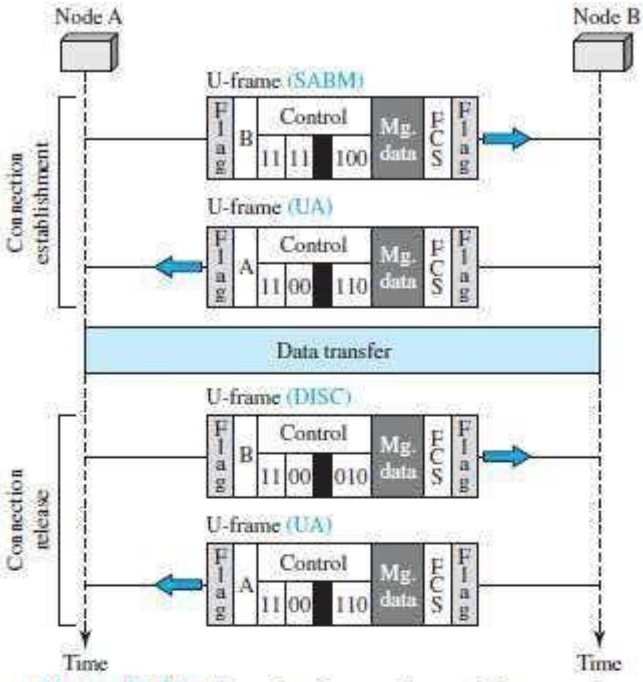

Figure 11.18 *Example of connection and disconnection*

3.9 POINT-TO-POINT PROTOCOL (PPP)

- PPP is one of the most common protocols for point-to-point access.
- Today, millions of Internet users who connect their home computers to the server of an ISP use PPP.

3.9.1 Framing

- PPP uses a character-oriented (or byte-oriented) frame (Figure 11.20).

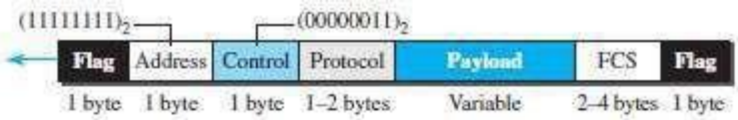

Figure 11.20 *PPP frame format*

- Various fields of PPP frame are:

1) Flag

- This field has a synchronization pattern 01111110.
- This field identifies both the beginning and the end of a frame.

2) Address

- This field is set to the constant value 11111111 (broadcast address).

3) Control

- This field is set to the constant value 00000011 (imitating unnumbered frames in HDLC). PPP does not provide any flow control.
- Error control is also limited to error detection.

4) Protocol

- This field defines what is being carried in the payload field.
- Payload field carries either i) user data or ii) other control information. By default, size of this field = 2 bytes.

5) Payload field

- This field carries either i) user data or ii) other control information.
- By default, maximum size of this field = 1500 bytes.

- This field is byte-stuffed if the flag-byte pattern appears in this field. Padding is needed if the payload-size is less than the maximum size. **6) FCS**
- This field is the PPP error-detection field.
- This field can contain either a 2- or 4-byte standard CRC.

3.9.1.1 Byte Stuffing

- Since PPP is a byte-oriented protocol, the flag in PPP is a byte that needs to be escaped whenever it appears in the data section of the frame.
- The escape byte is 01111101, which means that every time the flag like pattern appears in the data, this extra byte is stuffed to tell the receiver that the next byte is not a flag.
- Obviously, the escape byte itself should be stuffed with another escape byte.

3.9.2 Transition Phases

- The transition diagram starts with the dead state (Figure 11.21). **1) Dead State**

 In dead state, there is no active carrier and the line is quiet.

2) Establish State

 When 1 of the 2 nodes starts communication, the connection goes into the establish state. In establish state, options are negotiated between the two parties.

3) Authenticate State

 If the 2 parties agree that they need authentication,
Then the system needs to do authentication;
Otherwise, the parties can simply start communication.

4) Open State

 Data transfer takes place in the open state.

5) Terminate State

 When 1 of the endpoints wants to terminate connection, the system goes to terminate state.

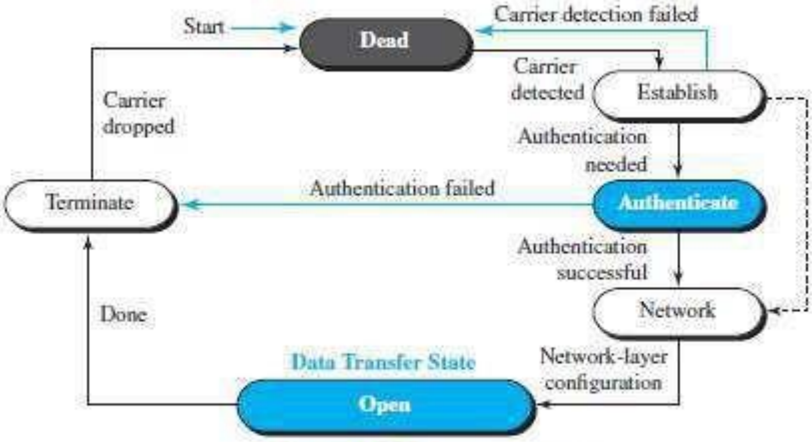

Figure 11.21 *Transition phases*

MODULE-WISE QUESTIONS

MODULE 3: ERROR-DETECTION AND CORRECTION

1) Explain two types of errors (4*)
2) Compare error detection vs. error correction (2)
3) Explain error detection using block coding technique. (10*)
4) Explain hamming distance for error detection (6*)
5) Explain parity-check code with block diagram. (6*)
6) Explain CRC with block diagram & an example. (10*)
7) Write short notes on polynomial codes. (5*)
8) Explain internet checksum algorithm along with an example. (6*)
9) Explain the following:
   i) Fletcher checksum and ii) Adler checksum (8)
10) Explain various FEC techniques. (6)

DATA LINK CONTROL

1) Explain two types of frames. (2)
2) Explain character oriented protocol. (6*)
3) Explain the concept of byte stuffing and unstuffing with example. (6*)
4) Explain bit oriented protocol. (6*)
5) Differentiate between character oriented and bit oriented format for Framing. (6*)
6) Compare flow control and error control. (4)
7) With a neat diagram, explain the design of the simplest protocol with no flow control. (6)
8) Write algorithm for sender site and receiver site for the simplest protocol. (6)
9) Explain Stop-and-Wait protocol (8*)
10) Explain the concept of Piggybacking (2*) 11) Explain in detail HDLC frame format. (8*)
12) Explain 3 type of frame used in HDLC (8*)
13) With a neat schematic, explain the frame structure of PPP protocol. (8*)
14) Explain framing and transition phases in Point-to-Point Protocol. (8*)