



# Chapter 11 Data Link Control

### Why Data Link Control?

The data link layer is the connection between a user workstation and the network.

For example, it is the connection between:

- your laptop and a wireless LAN
- your workstation and a wired LAN
- your computer at home and your ISP
- DePaul university and its ISP
- a business and ...

How does the data get from this workstation on the desk here to a server in, say, New York?

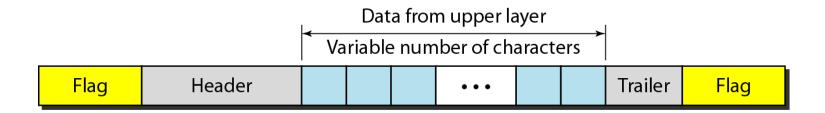
### 11-1 FRAMING

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.

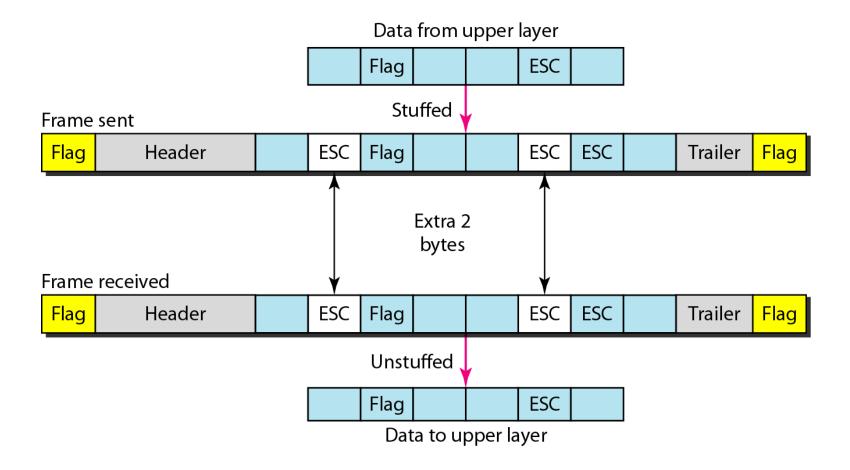
Topics discussed in this section:

Fixed-Size Framing Variable-Size Framing

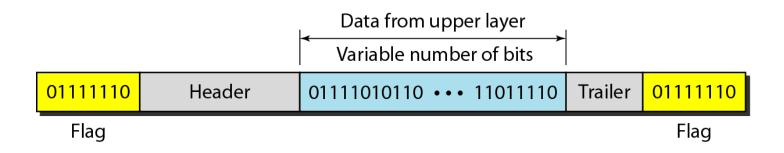
Figure 11.1 A frame in a character-oriented protocol



### Figure 11.2 Byte stuffing and unstuffing

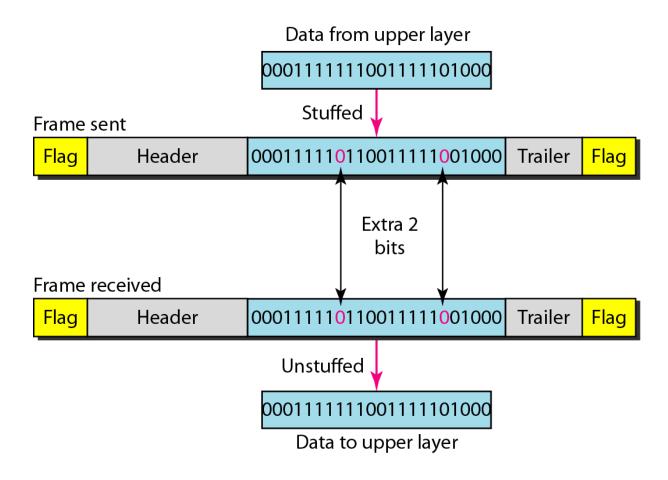


#### Figure 11.3 A frame in a bit-oriented protocol



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

## -

### Note

Flow control refers to a set of procedures used to restrict the amount of data that the sender can send before waiting for acknowledgment.

## •

### Note

Error control in the data link layer is based on automatic repeat request, which is the retransmission of data.

### **Error Control**

Once an error is detected, what is the receiver going to do?

- 1. Do nothing
- 2. Return an error message to the transmitter
- 3. Fix the error with no further help from the transmitter

## **Error Control - Do Nothing**

Seems like a strange way to control errors but some lower layer protocols such as frame relay perform this type of error control.

For example, if frame relay detects an error, it simply tosses the frame. No message is returned.

Frame relay assumes a higher protocol (such as TCP/IP) will detect the tossed frame and ask for retransmission

## Error Control - Return an Error Message

Once an error is detected, an error message is returned to the transmitter

Two basic forms:

**Stop-and-wait error control** 

**Sliding window error control** 

## **Stop-and-Wait Error Control**

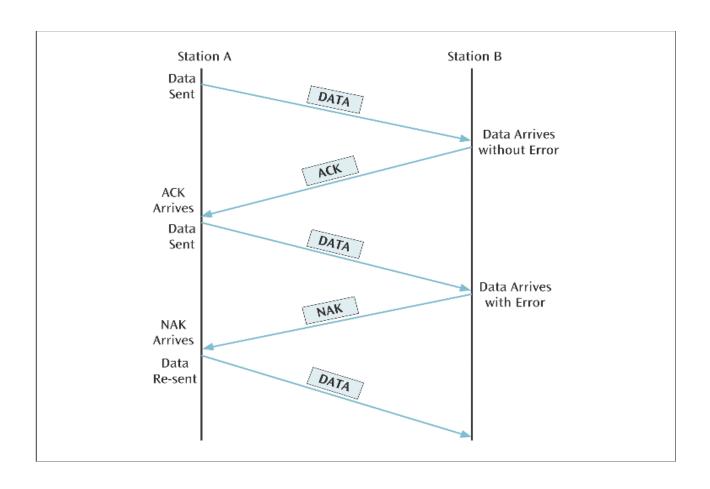
Stop-and-wait is the simplest of the error control protocols.

A transmitter sends a frame then stops and waits for an acknowledgment.

If a positive acknowledgment (ACK) is received, the next frame is sent.

If a negative acknowledgment (NAK) is received, the same frame is transmitted again.

Figure 6-8 Sample dialog using Stop-and-wait ARQ



# **Stop and Wait Link Utilization**

- $T = t_{prop} + t_{frame} + t_{proc} + t_{prop} + t_{ack} + t_{proc}$
- t<sub>proc</sub> and t<sub>ack</sub> negligible, so
- $T_{Total} = n(2t_{prop} + t_{frame}) \text{ for n frames}$
- $U_{\text{tilization}} = (n * t_{\text{frame}}) / n(2t_{\text{prop}} + t_{\text{frame}})$
- $= t_{frame} / (2t_{prop} + t_{frame})$
- With  $a = t_{prop} / t_{frame} U = 1 / (1 + 2a)$

# **Stop and Wait Link Utilization**

- Furthermore, a = t<sub>prop</sub> / t<sub>frame</sub>
- = Propagation Time / Frame Transmission Time
- = (D/V) / (L/R)
- Where:
  - D = distance of link
  - V = velocity of propagation (air = speed of light (3 x 10<sup>8</sup> m/s); fiber = same; copper = 0.67 x speed of light)
  - L = length of frame in bits
  - R = data rate in bps

# **Stop and Wait Link Utilization Example**

- Consider a WAN using ATM, 2 stations 1000 km apart, ATM frame size = 424 bits, standard data rate = 155.52 Mbps
- Frame Transmission Time (L/R) = $424/155.52 \times 10^6 = 2.7 \times 10^{-6}$  seconds
- Assume optical link:
- Propagation Time (D/V) = 10<sup>6</sup> m / 3 x 10<sup>8</sup> m/sec
- $= 0.33 \times 10^{-2} seconds$

# **Stop and Wait Link Utilization Example**

- Thus,  $a = 0.33 \times 10^{-2} / 2.7 \times 10^{-6} = 1222$
- U = 1/(1+2a) = 1/(1+2x1222) = 0.0004 ouch!
- Another example: A LAN
- $V = 2 \times 10^8 \text{ m/s}$
- L = 1000 bits
- R = 10 Mbps
- 11.19 D = 0.1 km = 100 m

# **Stop and Wait Link Utilization Example**

• Thus, a = (D/V) / (L/R) = 0.005

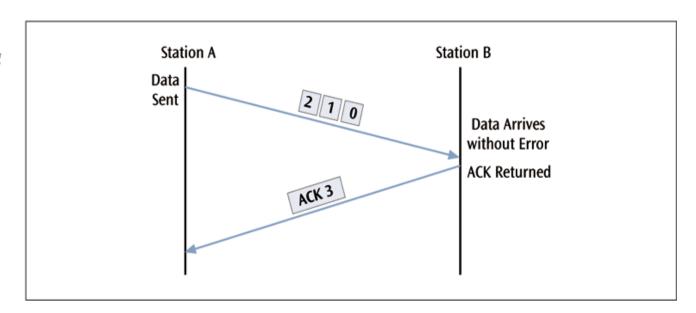
U = 1/(1+2a) = 0.99 No ouch!

These techniques assume that multiple frames are in transmission at one time

A sliding window protocol allows the transmitter to send up to the *window size* frames before receiving any acknowledgments.

When a receiver does acknowledge receipt, the returned ACK contains the number of the frame expected *next*.

Figure 6-9
Example of sliding window



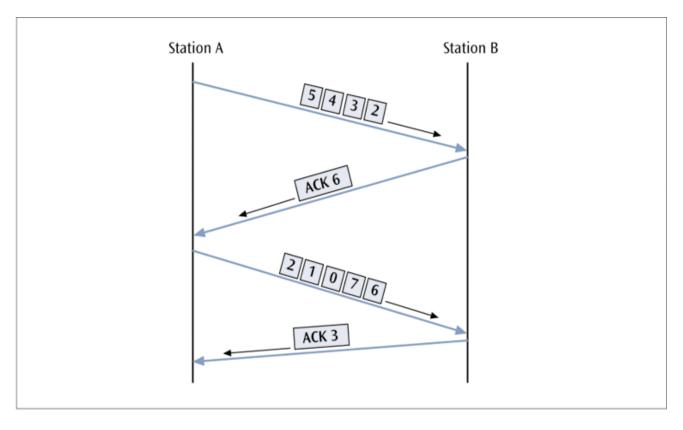
Older sliding window protocols numbered each frame or packet that was transmitted

More modern sliding window protocols number each byte within a frame

Let's look at an example in which the packets are numbered:

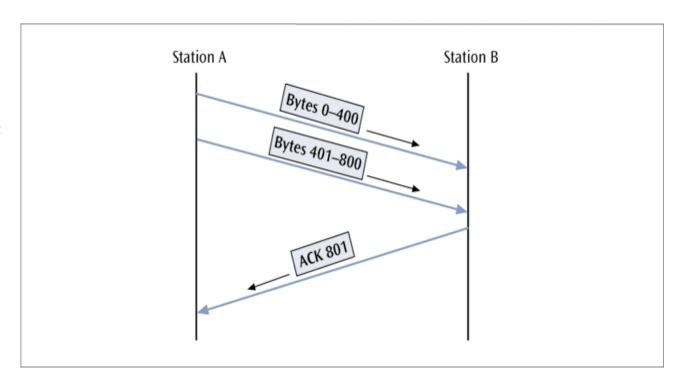
Figure 6-10

Normal transfer of data between two stations with numbering of the packets



#### This example shows each byte numbered

Figure 6-11
Normal transfer of
data between two
stations with
numbering of the bytes



Notice that an ACK is not always sent after each frame is received. It is more efficient to wait for a few received frames before returning an ACK.

How long should you wait until you return an ACK?

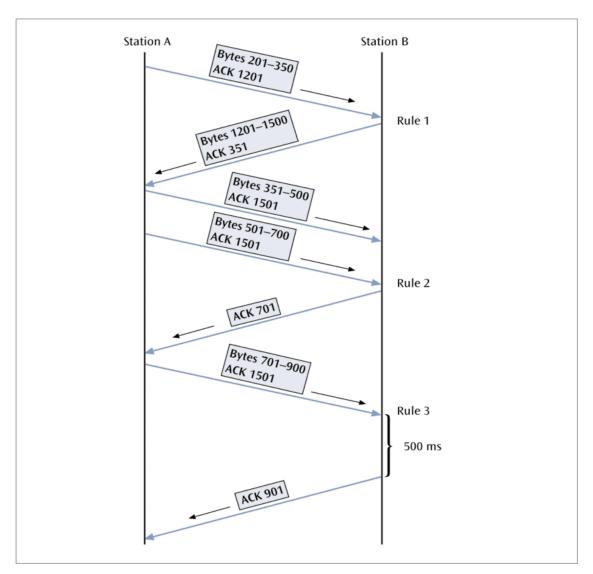
Using TCP/IP, there are some basic rules concerning ACKs

Rule 1: If a receiver just received data and wants to send its own data, piggyback an ACK along with that data

Rule 2: If a receiver has no data to return and has just ACKed the last packet, receiver waits 500 ms for another packet. If while waiting, another packet arrives, send the ACK immediately

Rule 3: If a receiver has no data to return and has just ACKed the last packet, receiver waits 500 ms. No packet, send ACK

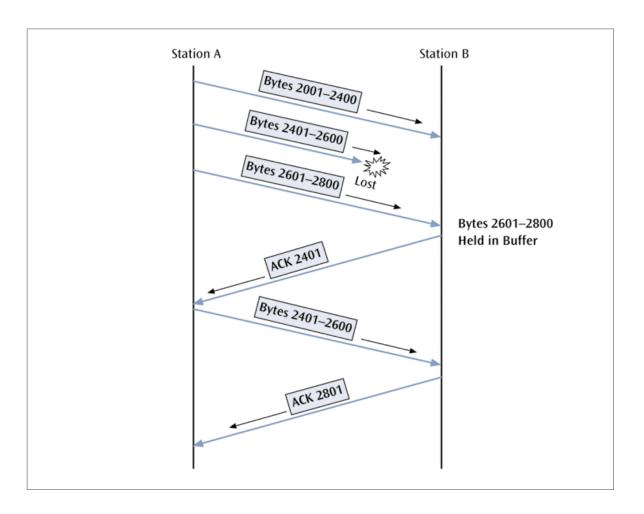
Figure 6-12
Three examples of returning an acknowledgment (ACK)



What happens when a packet is lost?

As shown in the next slide, if a frame is lost, the following frame will be "out of sequence". The receiver will hold the out of sequence bytes in a buffer and request the sender to retransmit the missing frame.

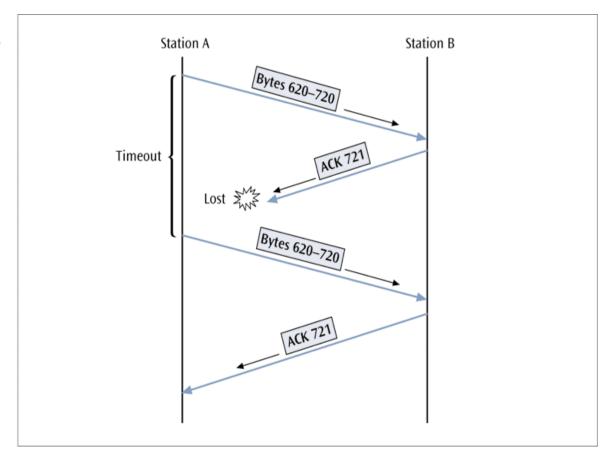
Figure 6-13 A lost packet and Station B's response



What happens when an ACK is lost?

As shown in the next slide, if an ACK is lost, the sender will wait for the ACK to arrive and eventually time-out. When the time-out occurs, the sender will resend the last frame.

Figure 6-14
A lost acknowledgment and the retransmission of a packet



# Sliding Window Performance

- U = 1 if W >= 2a + 1 where W = window size
- Thus, Utilization = 1 (100%) where ACK for frame 1 reaches A before A has exhausted its window
- U = W / (2a+1) if W < 2a + 1 (A's window closes)</p>

# Sliding Window Performance

- Example: What is U for a 1000-bit frame on a 1 Mbps satellite link with 270 ms delay with a window size of 127?
- a = Prop/Tran = .270
  sec/(1000/1000000)
- a = 270
- 2a + 1 = 541
- Is W < 2a + 1? Yes, so U = W / (2a + 1)
- U = 127 / 541 = 0.23

## **High Level Data Link Control**

- One of the more popular data link control protocols
- Similar to IBM's SDLC but more flexible
- Many data link protocols are based on HDLC – thus if you learn HDLC, you will understand many others, such as all the LAP standards

## **HDLC Station Types**

### Primary station

- Controls operation of link
- Frames issued are called commands
- Maintains separate logical link to each secondary station
- Secondary station
  - Under control of primary station
  - Frames issued called responses
- Combined station
  - May issue commands and responses

## **HDLC Link Configurations**

### Unbalanced

- One primary and one or more secondary stations
- Supports full duplex and half duplex
- Balanced
  - Two combined stations
  - Supports full duplex and half duplex

## **HDLC Transfer Modes (1)**

- Normal Response Mode (NRM)
  - Unbalanced configuration
  - Primary initiates transfer to secondary
  - Secondary may only transmit data in response to command from primary
  - Used on multi-drop lines
  - Host computer as primary
  - Terminals as secondary

## **HDLC Transfer Modes (2)**

- Asynchronous Balanced Mode (ABM)
  - Balanced configuration
  - Either station may initiate transmission without receiving permission
  - Most widely used
  - No polling overhead

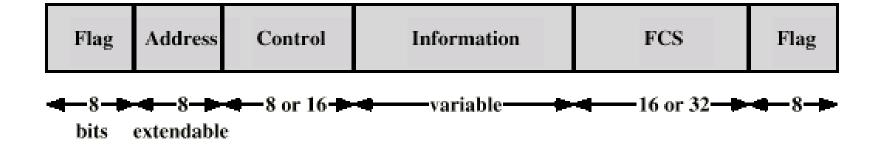
## **HDLC Transfer Modes (3)**

- Asynchronous Response Mode (ARM)
  - Unbalanced configuration
  - Secondary may initiate transmission without permission from primary
  - Primary responsible for line
  - Rarely used

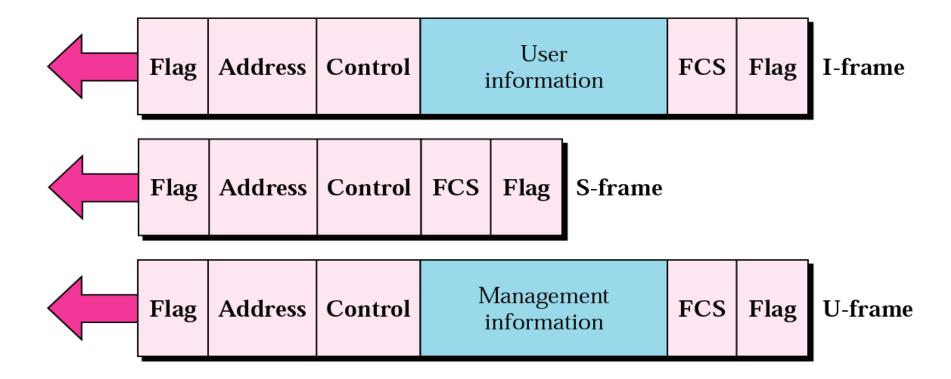
### **Frame Structure**

- Synchronous transmission
- All transmissions in frames
- Single frame format for all data and control exchanges

## **Frame Structure Diagram**

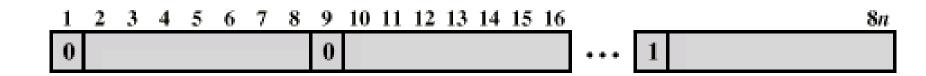






## **Address Field**

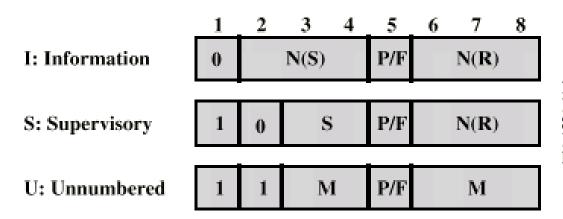
- Identifies secondary station that sent or will receive frame
- Usually 8 bits long
- May be extended to multiples of 7 bits
  - LSB of each octet indicates that it is the last octet (1) or not (0)
- All ones (11111111) is broadcast



## **Control Field**

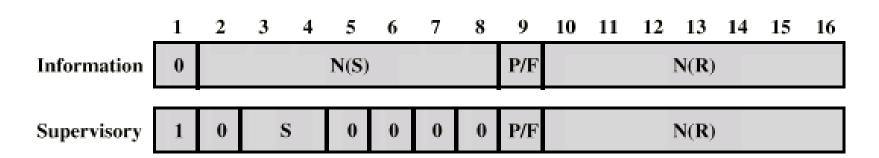
- Different for different frame type
  - Information data to be transmitted to user (next layer up)
    - Flow and error control piggybacked on information frames
  - Supervisory ARQ when piggyback not used
  - Unnumbered supplementary link control
- First one or two bits of control field identify frame type

## **Control Field Diagram**



N(S) = Send sequence number N(R) = Receive sequence number S = Supervisory function bits M = Unnumbered function bits P/F = Poll/final bit

(c) 8-bit control field format



(d) 16-bit control field format

# HDLC Commands and Responses

### **Supervisory**

Receive Ready (RR) - positive acknowledgment

**Receive Not Ready (RNR)** 

Reject (REJ)

**Selective Reject (SREJ)** 

### Unnumbered

Set normal response/extended mode (SNRM/SNRME)

Set asynchronous response/extended mode (SARM/SARME)

Set asynchronous balanced/extended mode (SABM/SABME)

Set initialization mode (SIM)

**Disconnect (DISC)** 

**Unnumbered ack (UA)** 

**Disconnected mode (DM)** 

Request disconnect (RD)

Request initialization mode (RIM)

**Unnumbered information (UI)** 

**Unnumbered poll (UP)** 

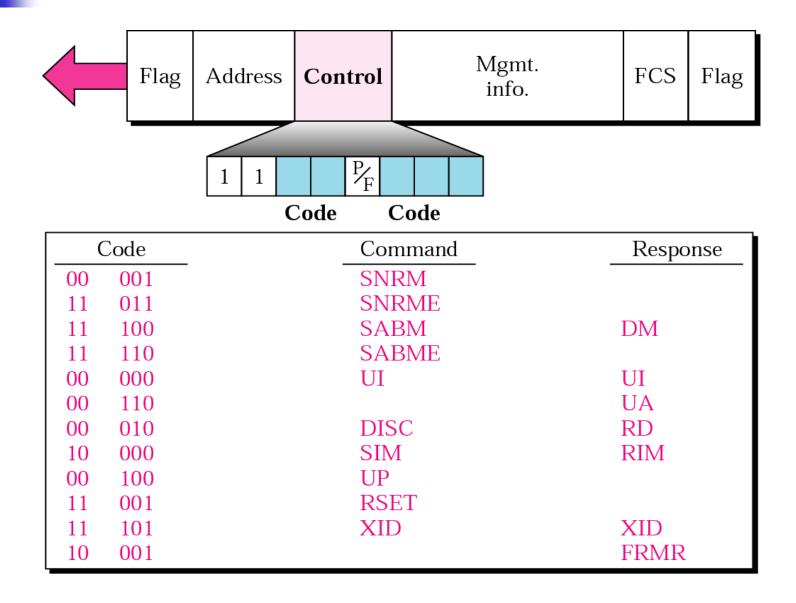
**Reset (RSET)** 

**Exchange identification (XID)** 

Test (TEST)

Frame reject (FRMR)

### U-frame control field in HDLC



## **Poll/Final Bit**

- Use depends on context
- Command frame
  - P bit
  - 1 to solicit (poll) response from peer
- Response frame
  - F bit
  - 1 indicates response to soliciting command

## **Information Field**

- Only in information and some unnumbered frames
- Must contain integral number of octets
- Variable length

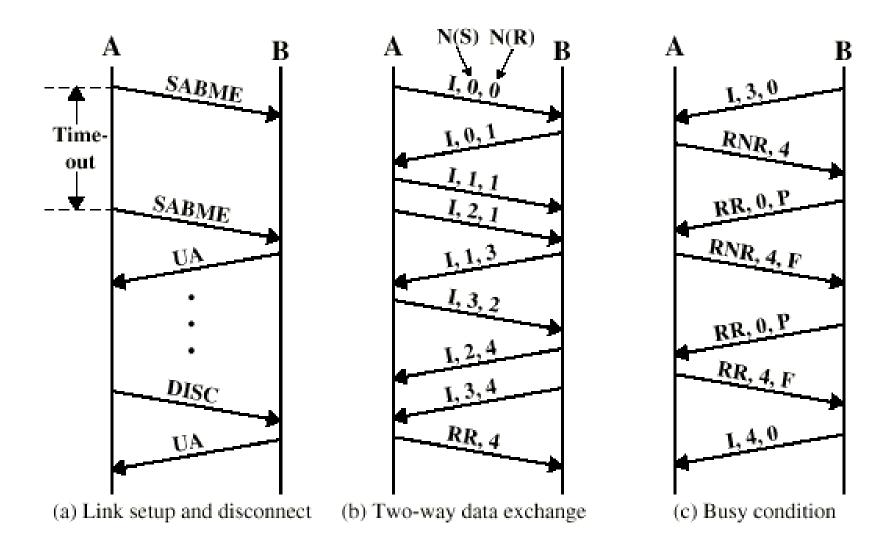
## Frame Check Sequence Field

- FCS
- Error detection
- 16 bit CRC
- Optional 32 bit CRC

## **HDLC Operation**

- Exchange of information, supervisory and unnumbered frames
- Three phases
  - Initialization
  - Data transfer
  - Disconnect

## Examples of Operation (1)



## **Examples of Operation (2)**

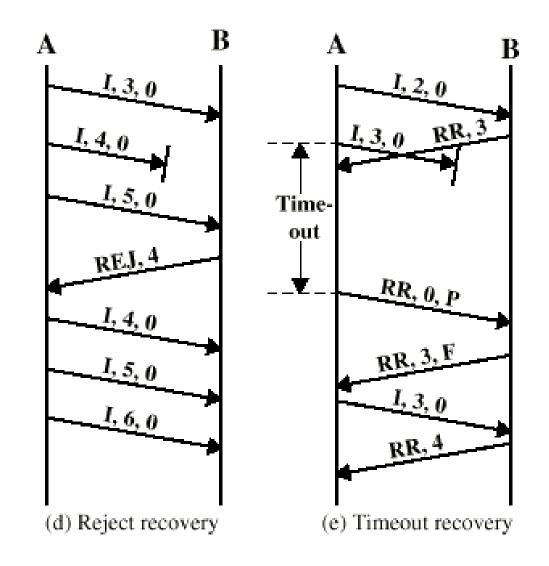
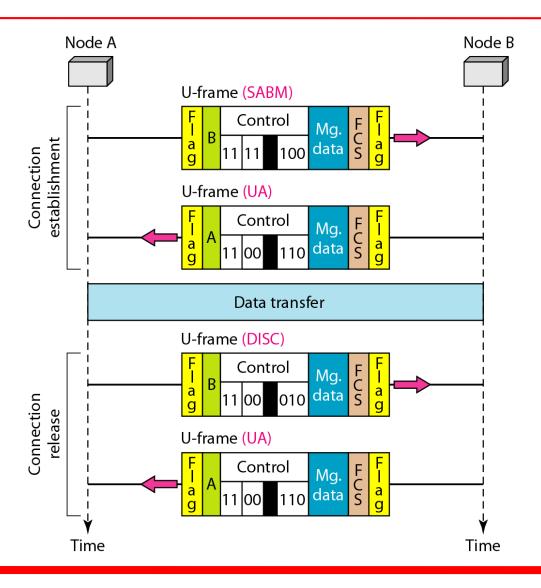


Figure 11.29 Example of connection and disconnection



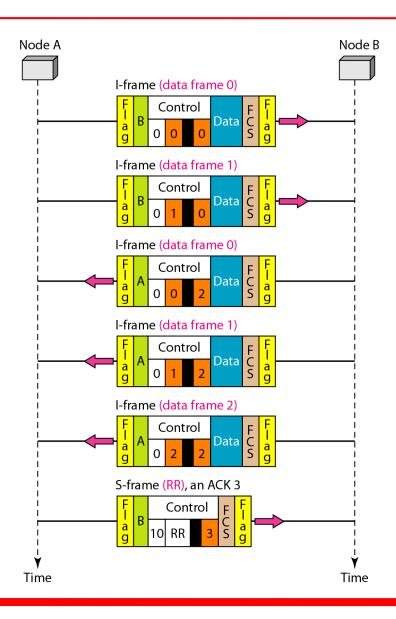
## Example 11.10

Figure 11.30 shows an exchange using piggybacking. Node begins the exchange of information with I-frame numbered 0 followed by another I-frame numbered 1. Node B piggybacks its acknowledgment of both frames onto an I-frame of its own. Node B's first I-frame is also numbered 0 [N(S) field] and contains a 2 in its N(R) field, acknowledging the receipt of A's frames 1 and 0 and indicating that it expects frame 2 to arrive next. Node B transmits its second and third I-frames (numbered 1 and 2) before accepting further frames from node A.

## Example 11.10 (continued)

Its N(R) information, therefore, has not changed: B frames 1 and 2 indicate that node B is still expecting A's frame 2 to arrive next. Node A has sent all its data. Therefore, it cannot piggyback an acknowledgment onto an I-frame and sends an S-frame instead. The RR code indicates that A is still ready to receive. The number 3 in the N(R) field tells B that frames 0, 1, and 2 have all been accepted and that A is now expecting frame number 3.

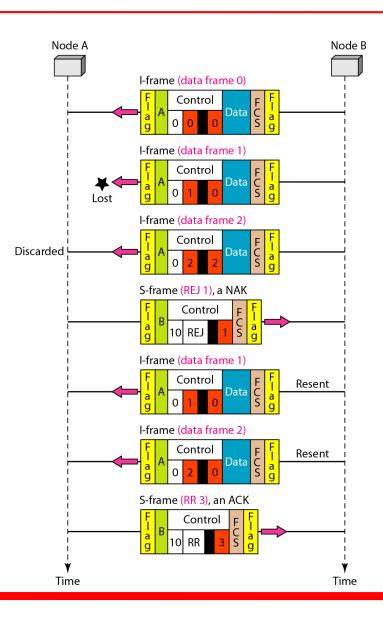
### Figure 11.30 Example of piggybacking without error



## Example 11.11

Figure 11.31 shows an exchange in which a frame is lost. Node B sends three data frames (0, 1, and 2), but frame 1 is lost. When node A receives frame 2, it discards it and sends a REJ frame for frame 1. Note that the protocol being used is Go-Back-N with the special use of an REJ frame as a NAK frame. The NAK frame does two things here: It confirms the receipt of frame 0 and declares that frame 1 and any following frames must be resent. Node B, after receiving the REJ frame, resends frames 1 and 2. Node A acknowledges the receipt by sending an RR frame (ACK) with acknowledgment number 3.

### Figure 11.31 Example of piggybacking with error



# Other DLC Protocols (LAPB,LAPD)

- Link Access Procedure, Balanced (LAPB)
  - Part of X.25 (ITU-T)
  - Subset of HDLC ABM
  - Point to point link between system and packet switching network node
- Link Access Procedure, D-Channel
  - ISDN (ITU-D)
  - ABM
  - Always 7-bit sequence numbers (no 3-bit)
  - 16 bit address field contains two subaddresses
    - One for device and one for user (next layer up)

## Other DLC Protocols (LLC)

- Logical Link Control (LLC)
  - IEEE 802
  - Different frame format
  - Link control split between medium access layer (MAC) and LLC (on top of MAC)
  - No primary and secondary all stations are peers
  - Two addresses needed
    - Sender and receiver
  - Error detection at MAC layer
    - 32 bit CRC
  - Destination and source access points (DSAP, SSAD)

# Other DLC Protocols (Frame Relay) (1)

- Streamlined capability over high speed packet witched networks
- Used in place of X.25
- Uses Link Access Procedure for Frame-Mode Bearer Services (LAPF)
- Two protocols
  - Control similar to HDLC
  - Core subset of control

# Other DLC Protocols (Frame Relay) (2)

- ABM
- 7-bit sequence numbers
- 16 bit CRC
- 2, 3 or 4 octet address field
  - Data link connection identifier (DLCI)
  - Identifies logical connection
- More on frame relay later

## Other DLC Protocols (ATM)

- Asynchronous Transfer Mode
- Streamlined capability across high speed networks
- Not HDLC based
- Frame format called "cell"
- Fixed 53 octet (424 bit)
- Details later

## **Data Communications**

### Point-to-Point Protocol (PPP)

## **Basics of PPP**

- Used on point-to-point links such as modem dialup, DSL, and cable modem
- SLIP (serial line Internet protocol) was first but could only support IP and only static IP address assignment
- PPP solves both above problems

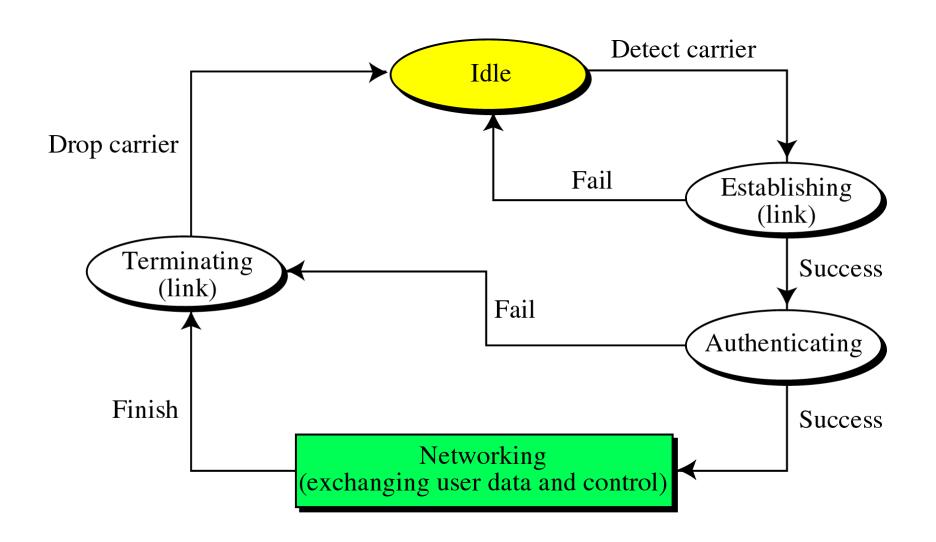
## **PPP States**

- Idle state link is not being used
- Establishing state one endpoint starts a dialog; options are exchanged between endpoints; several packets may be exchanged
- Authenticating state optional, two sides agree to authenticate (described later)

## **PPP States**

- Networking state primary state, exchange of user control and data packets can now be performed
- Terminate state one side wishes to tear down connection; several packets exchanged (housekeeping)
- (See figure next slide)

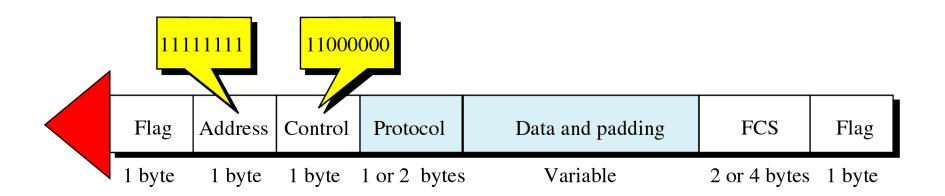
### **Transition States**



## **PPP Layers**

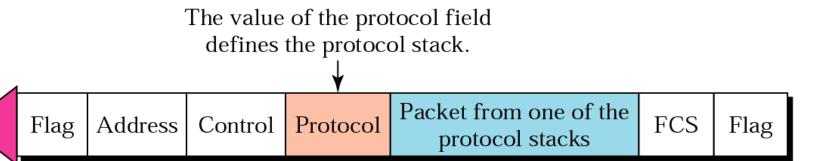
- PPP has only two layers physical and data link
- Physical layer is not defined it is whatever the user uses
- Data link layer looks like HDLC, except address field = 11111111 (broadcast), control field = 11000000 (a HDLC U-frame)

### **PPP Frames**



# **PPP Layers**

 The Data field carries the packets from one of three other protocols -Link Control Protocol, authentication protocols, and Network Control Protocol, all described shortly



# Link Control Protocol (LCP)

- Responsible for establishing, maintaining, configuring, terminating link, and negotiation
- All LCP packets are carried in payload field of PPP frame – PPP field Protocol = hex C021

## LCP Packet Encapsulated in a Frame

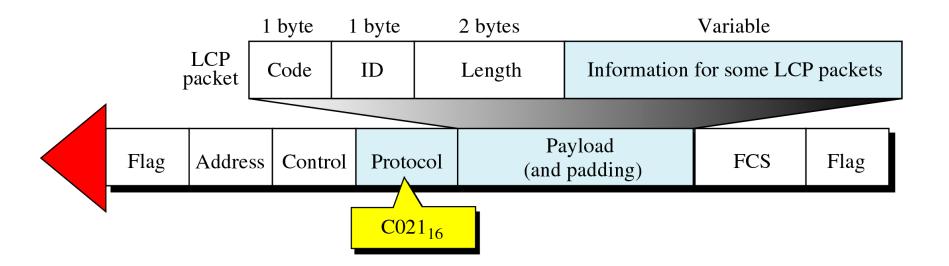


Table 12.1 LCP packets and their codes

	Code	Packet Type	Description
	01 <sub>16</sub>	Configure- request	Contains the list of proposed options and their values
	02 <sub>16</sub>	Configure-ack	Accepts all options proposed
	03 <sub>16</sub>	Configure-nak	Announces that some options are not acceptable
	04 <sub>16</sub>	Configure-reject	Announces that some options are not recognized
	05 <sub>16</sub>	Terminate- request	Requests to shut down the line
	06 <sub>16</sub>	Terminate-ack	Accepts the shut down request
	07 <sub>16</sub>	Code-reject	Announces an unknown code
	08 <sub>16</sub>	Protocol-reject	Announces an unknown protocol
	09 <sub>16</sub>	Echo-request	A type of hello message to check if the other end is alive
	0A <sub>16</sub>	Echo-reply	The response to the echo-request message
1	0B <sub>16</sub>	Discard-request	A request to discard the packet

#### Table 12.2 Common options

Option	Default
Maximum receive unit	1500
Authentication protocol	None
Protocol field compression	Off
Address and control field compression	Off

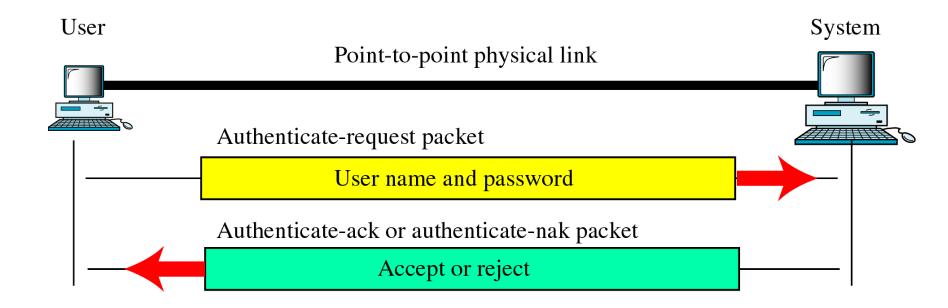
# **Authentication**

- Potentially important since this is dial-up communication
- Two possible protocols for authentication:
  - Password Authentication Protocol (PAP
  - Challenge Handshake Authentication Protocol (CHAP)

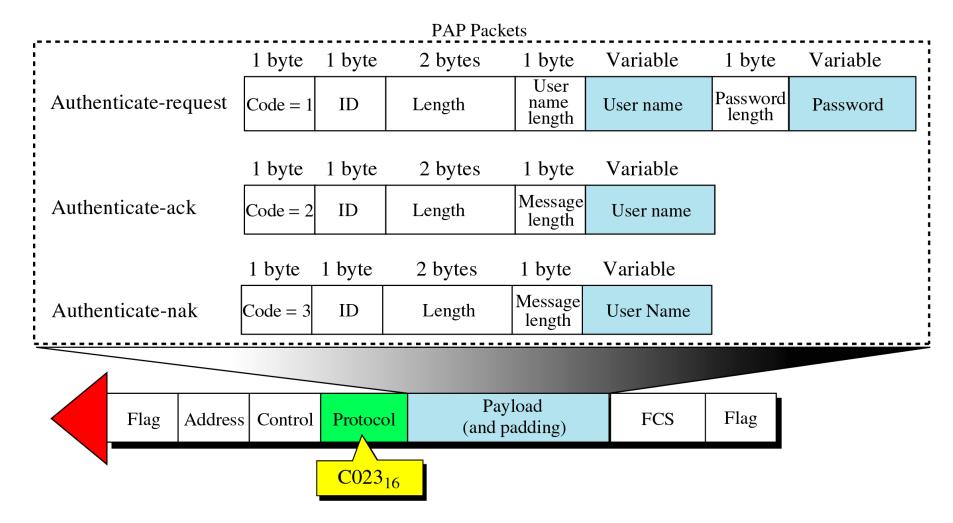
## **Authentication — PAP**

- Two-step process
  - User sends ID and password
  - System verifies
- PAP packets are encapsulate in a PPP frame
- There are 3 types of PAP packets (see the next two slides)

#### PAP



#### **PAP Packets**

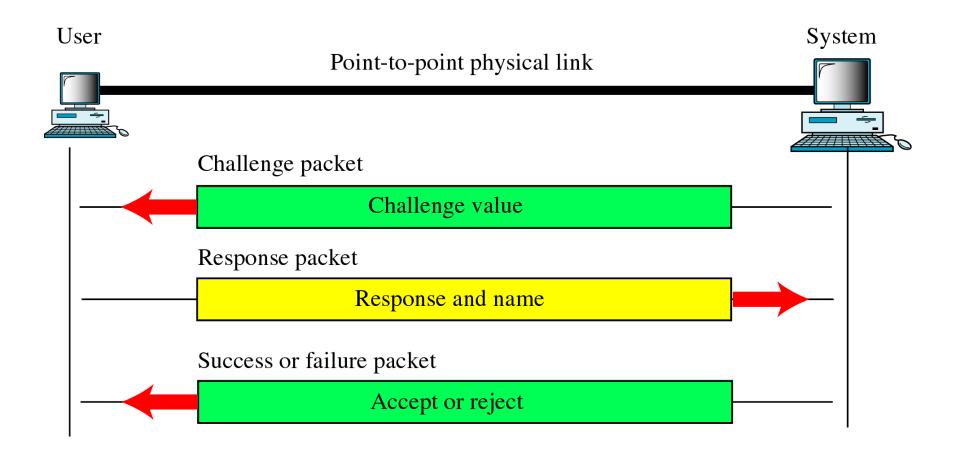


11.82

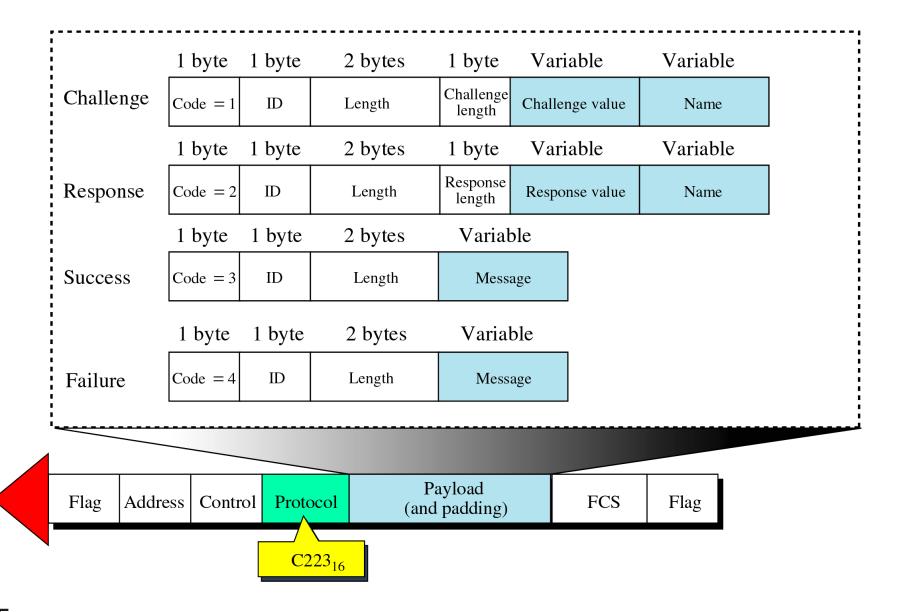
# **Authentication – CHAP**

- Three-way handshake
  - System sends a challenge packet
  - User applies a predefined function that takes the challenge value and the user's own password and creates a result
  - System does the same; then compares its result to user's result

#### **CHAP**



#### **CHAP Packets**



11.85

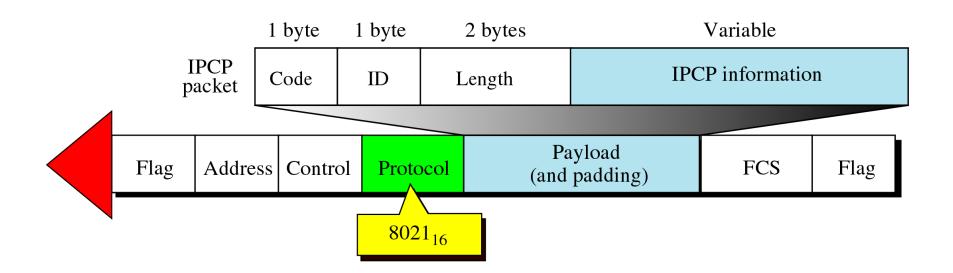
# **IPCP (An NCP Protocol)**

- Now that a link has been established and optional security has been established, we need to establish a network layer connection
- IPCP, or Internetwork Protocol Control Protocol, is an NCP (Network Control Protocol)

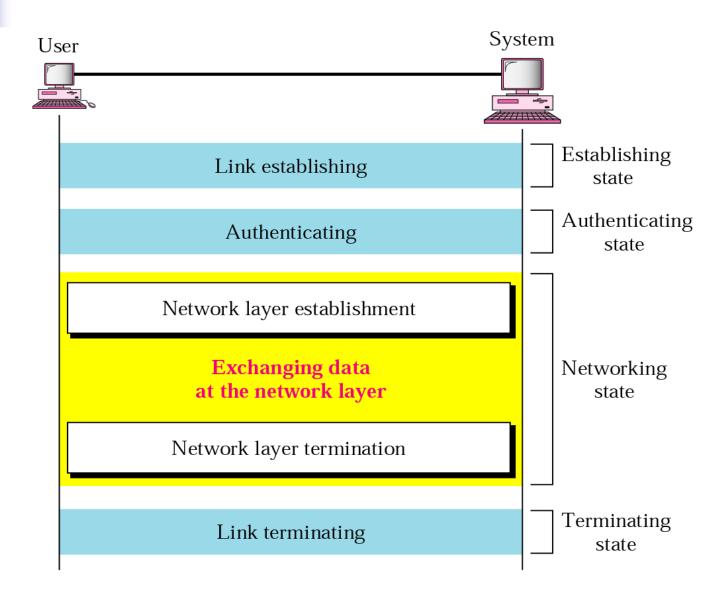
## **IPCP**

- Seven packet types:
  - Configure-request (01)
  - Configure-ACK (02)
  - Configure-NAK (03)
  - Configure-reject (04)
  - Terminate-request (05)
  - Terminate-ACK (06)
  - Code-reject (07)

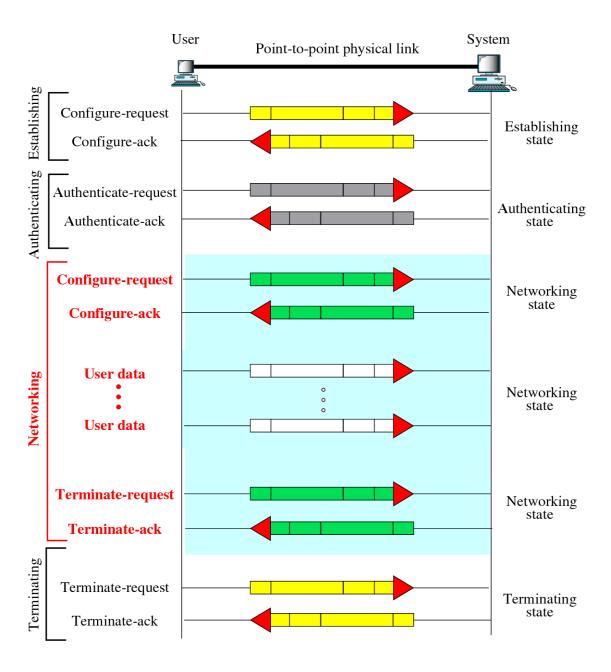
### IPCP Packet Encapsulated in PPP Frame



#### 12.10 An example



## An Example



# **Review Questions**

- 1. What is the utilization of stop and wait flow control? (Two stations 20 km apart, 1000 byte frames, 256 Kbps, UTP)
- What is the utilization of a sliding window system where stations are 100 km apart, 500 byte frame, 1 Mbps, microwave, window size = 255?
- 3. Why is the window size always 2<sup>n</sup> 1?
- 4. What are differences between go-back-n and selective reject?

# **Review Questions**

- 5. Ten frames sent, 5<sup>th</sup> frame is lost. What happens with go-back-N? With selective reject?
- 6. What is normal response mode in HDLC?
- 7. How does bit stuffing work?
- 8. What is the difference between HDLC and SDLC?
- 9. Show a sequence of PPP packet exchanges