# High-level Data Link Control (HDLC) and Point to Point Protocol (PPP)

Sachin Gajjar
sachin.gajjar@nirmauni.ac.in

# Reading Material for this topic

- DATA COMMUNICATIONS AND NETWORKING, Fourth Edition by Behrouz A. Forouzan, Tata McGraw-Hill
  - Chapter 11, Topic 11.6, 11.7

# High-level Data Link Control

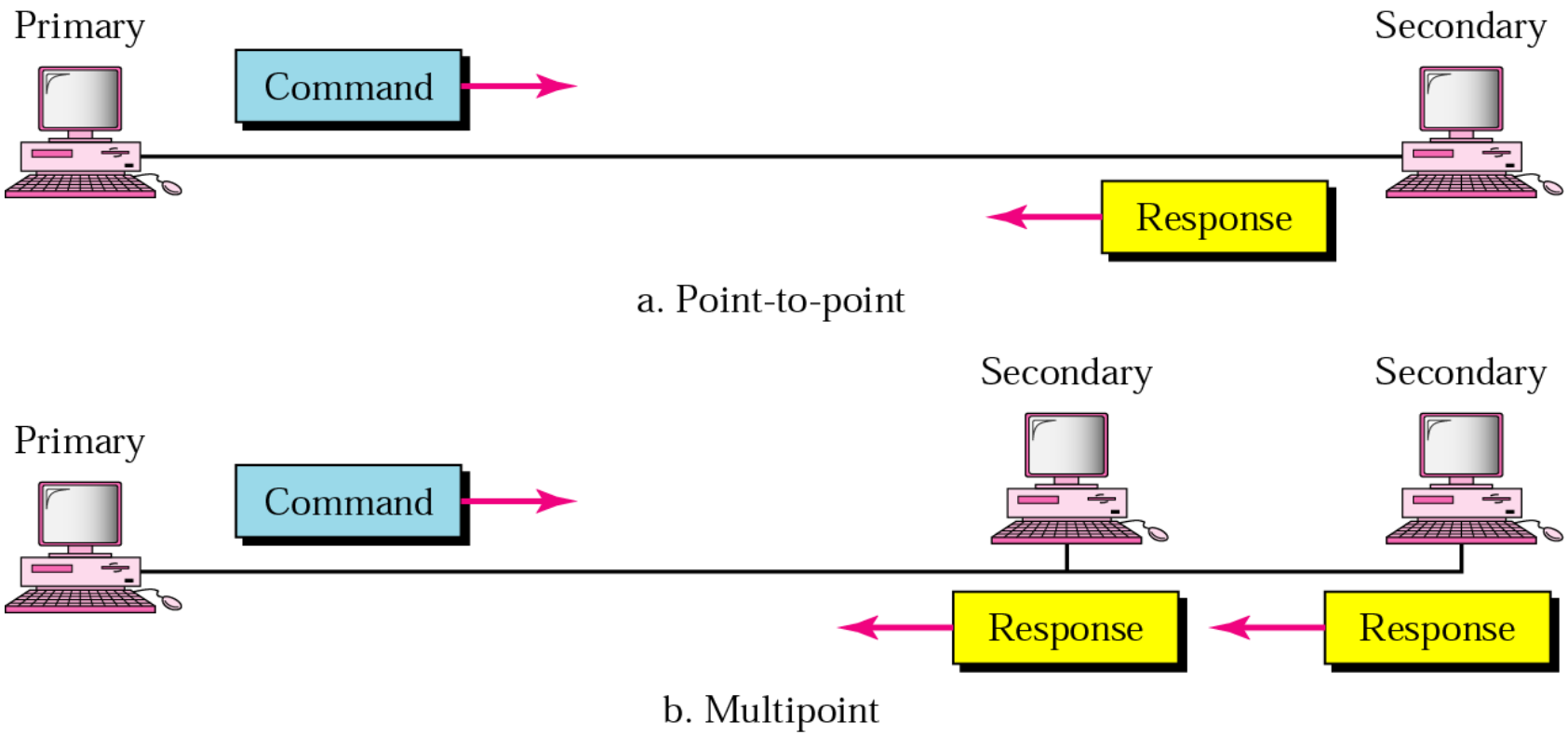
- A bit-oriented protocol for communication over point-to-point and multipoint links
- Implements ARQ mechanisms

# Configurations and Transfer Modes

- Two common transfer modes
- Normal Response Mode (NRM)
  - one  primary station and multiple secondary stations
  - primary station can send commands
  - secondary station can only respond

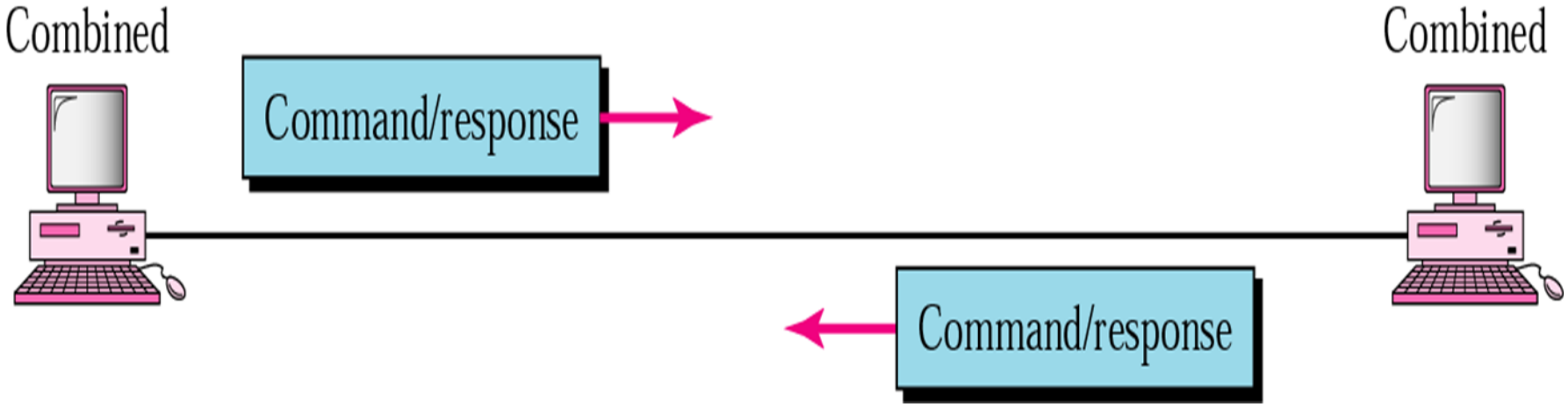
# NRM

- Used for both point-to-point and multiple-point links



# Asynchronous Balanced Mode (ABM)

- Configuration is balanced
- Link is point-to-point
- Each station can function as a primary and a secondary (acting as peers)
- This is the common mode today

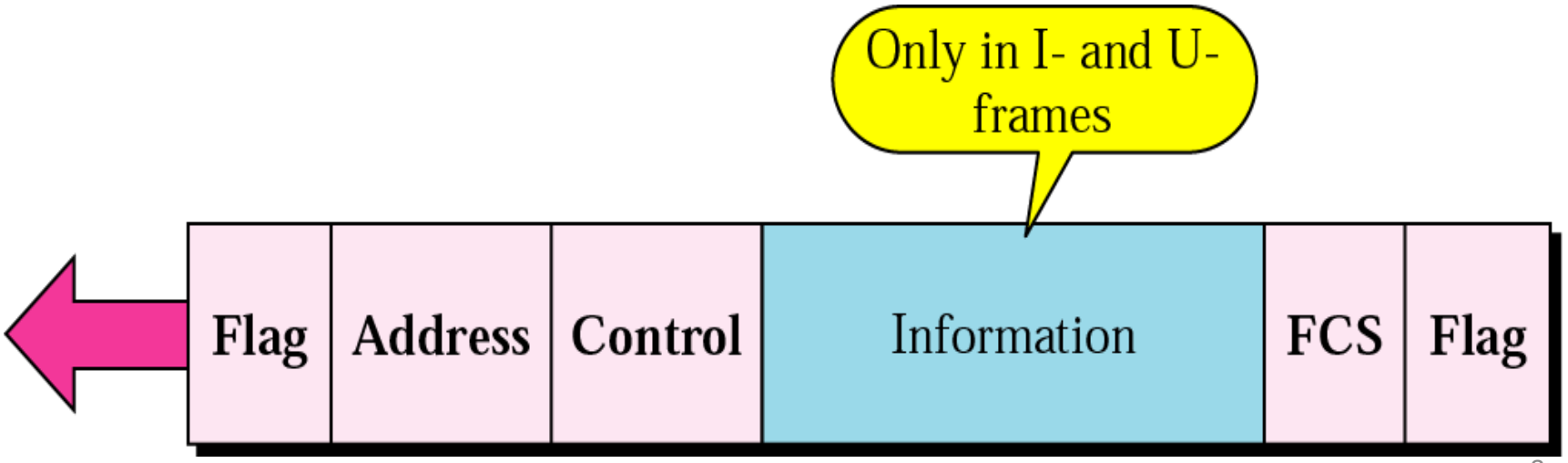


# Frames

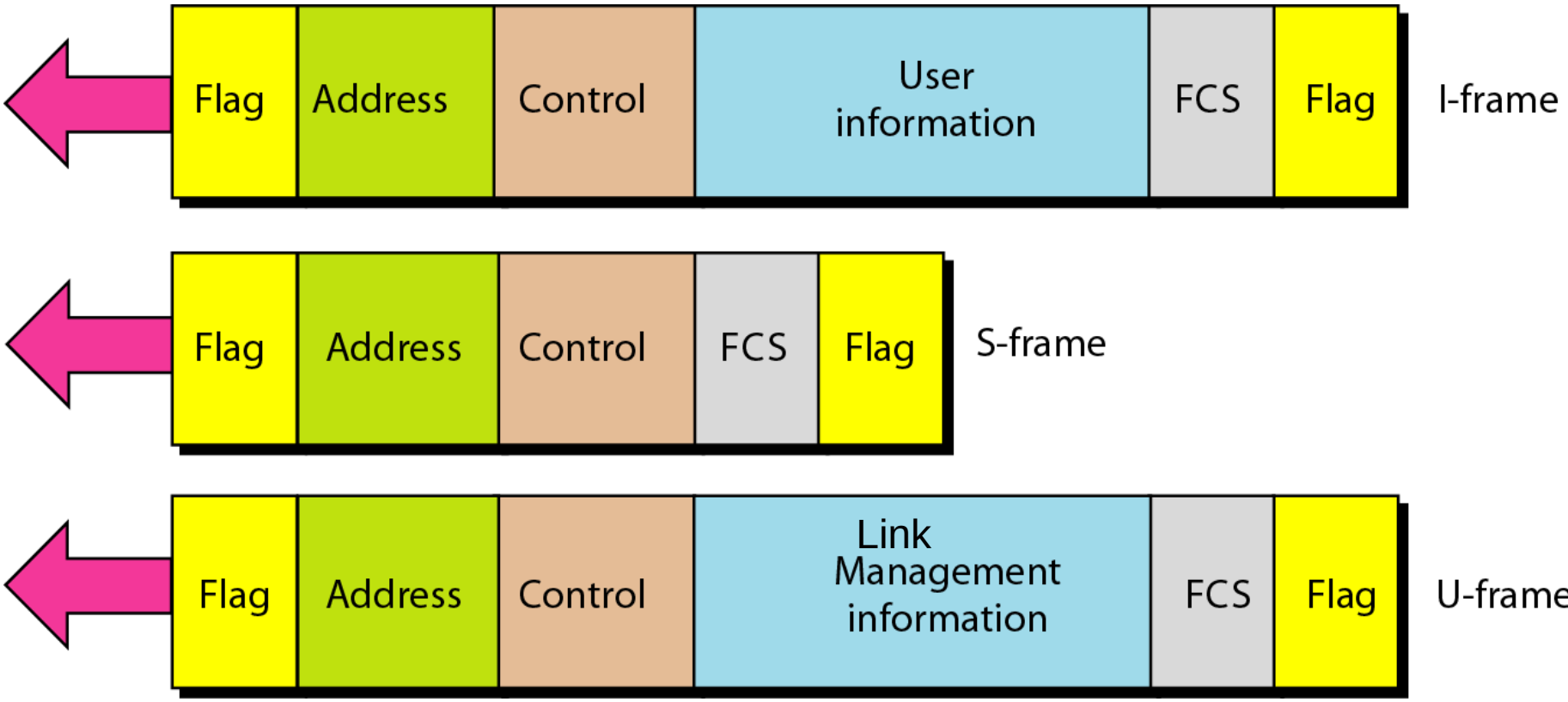
- I-frames  to transport user data and control information relating to user data (piggybacking).
- S-frames are used to transport control information
- U-frames are reserved for system management, for managing the link itself

# Frame Format

- Each frame may contain up to six fields
  - start flag, address, control, information, frame check sequence, end flag
- In multiple-frame transmissions, end flag of one frame can serve as start flag of next frame



# HDLC frames


Flag
Address
Control
User information
FCS
Flag
I-frame
Flag
Address
Control
FCS
Flag
S-frame
Flag
Address
Control
Link Management information
FCS
Flag
U-frame

# Fields

- Flag field = 01111110 = start/end of frame = synchronization pattern for receiver
- Address field = if primary station created frame, it contains “to address”, if secondary creates it contains “from address”
  - Length depends on needs of network
  - If address field is 1 byte, last bit = 1, 128 stations
  - If address is > 1 byte, all intermediate bytes will end with 0, the last will end with 1
  - Ending each intermediate byte with 0 indicates to receiver that there are more address bytes to come

# Fields

- Information field - contains user's data from network layer or management information
  - Its length can vary from one network to another.
- FCS field - frame check sequence is HDLC error detection field can contain 2/4-byte ITU-T CRC

# CRC

- Convert message to single binary word M
- Check word r = remainder of (M/k)

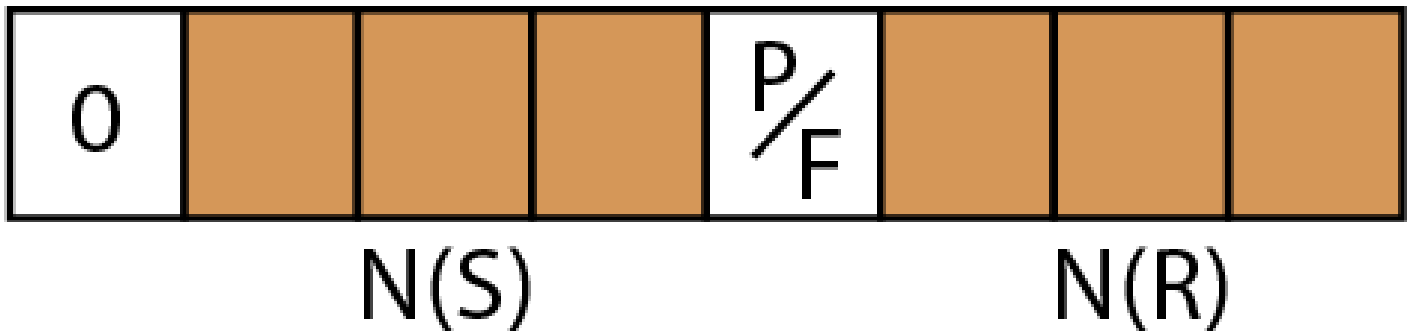
  k is known to both Tx/Rx
- Transmitter sends both message M and r

- Rx checks data by repeating calculation, dividing M by key word k, and verifying that the remainder is r
- If r=0, data is correct, if r!=0, data is corrupted

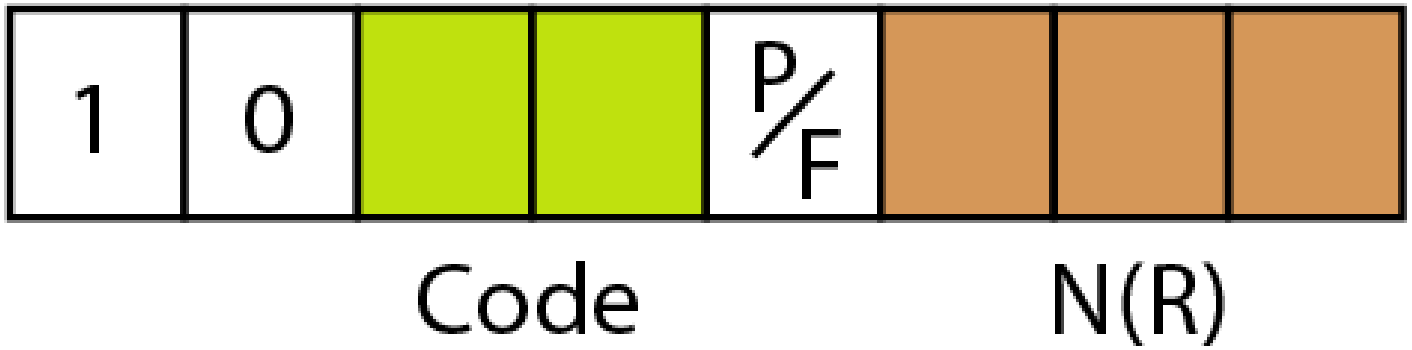
# Fields

- Control field is 1/2 byte segment of frame used for flow/error control
- Interpretation of bits field depends on frame type

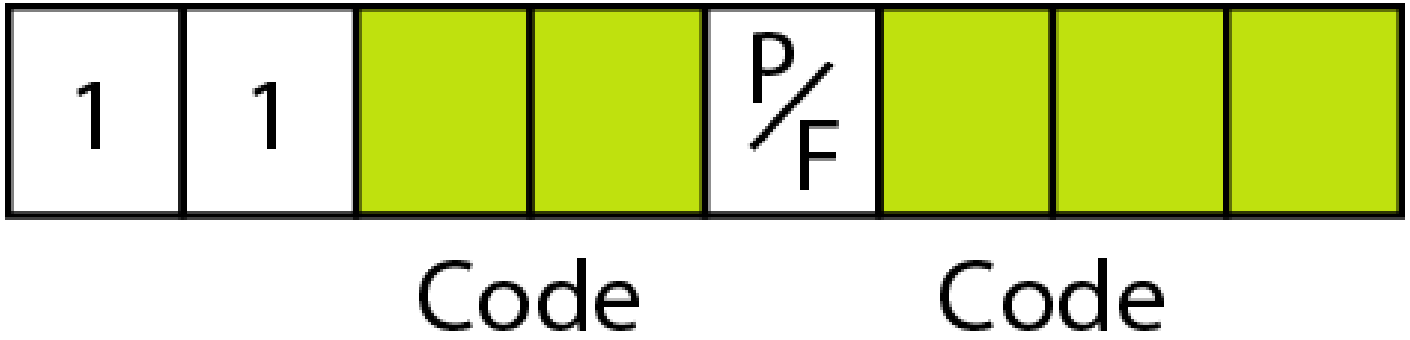
# Control field format for different frame types




I-frame



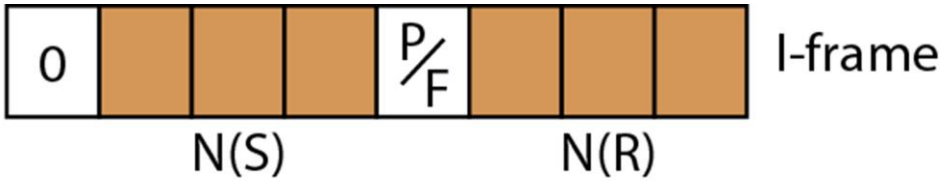

S-frame



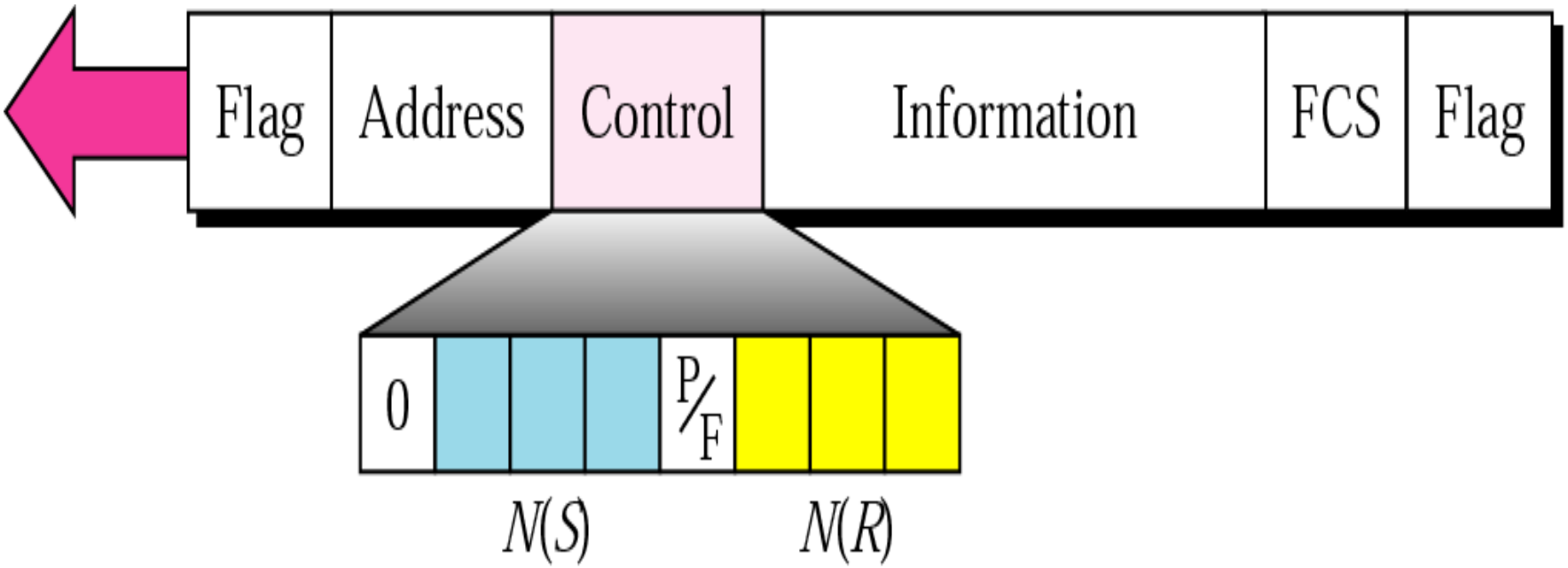

U-frame

# Control Field for I-Frames

- I-frames carry user data from network layer
- Can include flow/error control information
- Subfields in control field define these functions
- First bit of control field = 0, means frame = I-frame.
- Next 3 bits = N(S) = sequence number of frame
- Last 3 bits = N(R) = ack no. when piggybacking is used
- Single bit b/w N(S) and N(R) = P/F bit
- When it is set (bit = 1) it can mean poll or final
- Is poll, when the frame sent by primary to secondary
- Is final, when the frame sent by secondary to primary

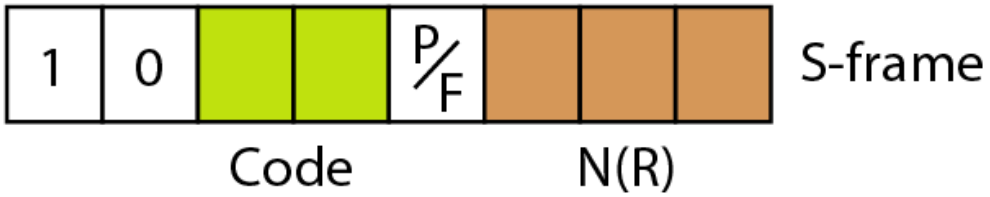


# I-frame



# Control Field for S-Frames

- Supervisory frames are used for flow and error control whenever piggybacking (data+ack) is either impossible or inappropriate
- S frames do not have information fields
- first 2 bits of control field = 10, means S-frame.
- last 3 bits = N(R) = ACK number,  NAK depending on type of S-frame



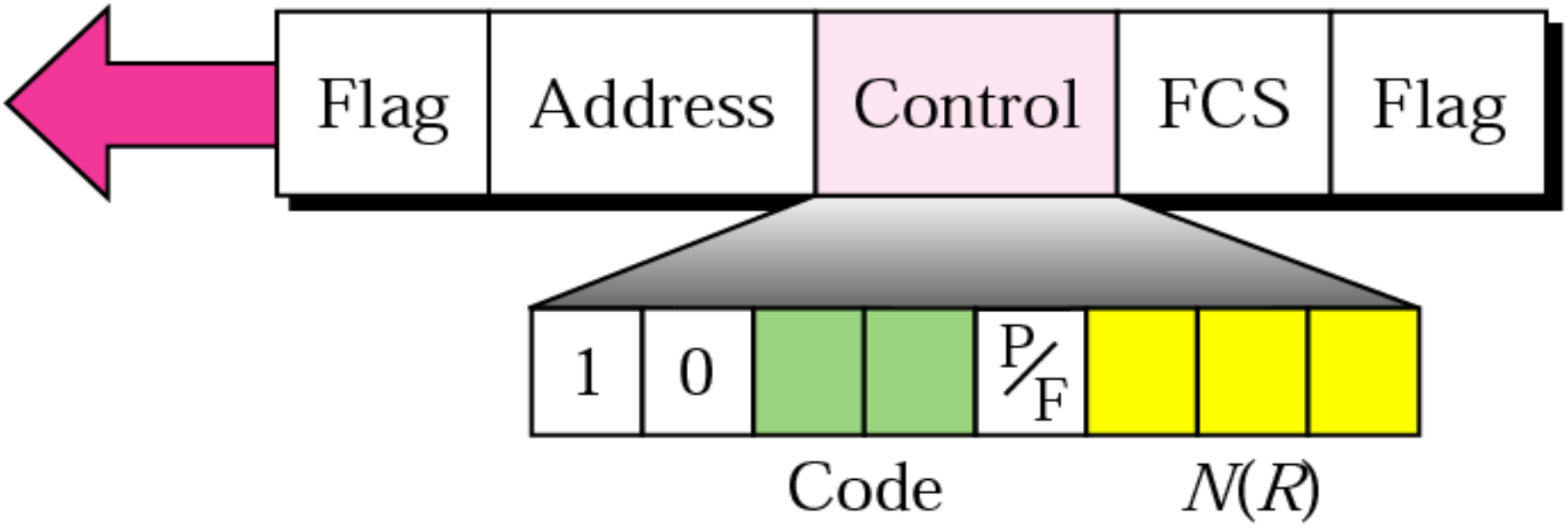
# Four types of S-frames

- 2 bits = code is to define type of S-frame itself
- Receive ready RR S-frame (00)
  - Acknowledges receipt of correct frame/group of frames
  - N(R) = ACK number
- Receive not ready RNR S-frame (10)
  - RR frame with additional function
  - ACKs receipt of frame/group of frames, announces that rx is busy, cannot receive more frames
  - acts as a kind of congestion control mechanism by asking the sender to slow down
  - N(R) = ACK number

# Four types of S-frames

- Reject REJ S-frame (01)
  - A NAK frame that can be used in Go-Back-N ARQ
  - informing sender, before sender time expires, that the last frame is lost or damaged
  - N(R) = negative ACK number
- Selective reject SREJ S-frame (11)
  - A NAK frame used in Selective Repeat ARQ
  - N(R) = negative  ACK number

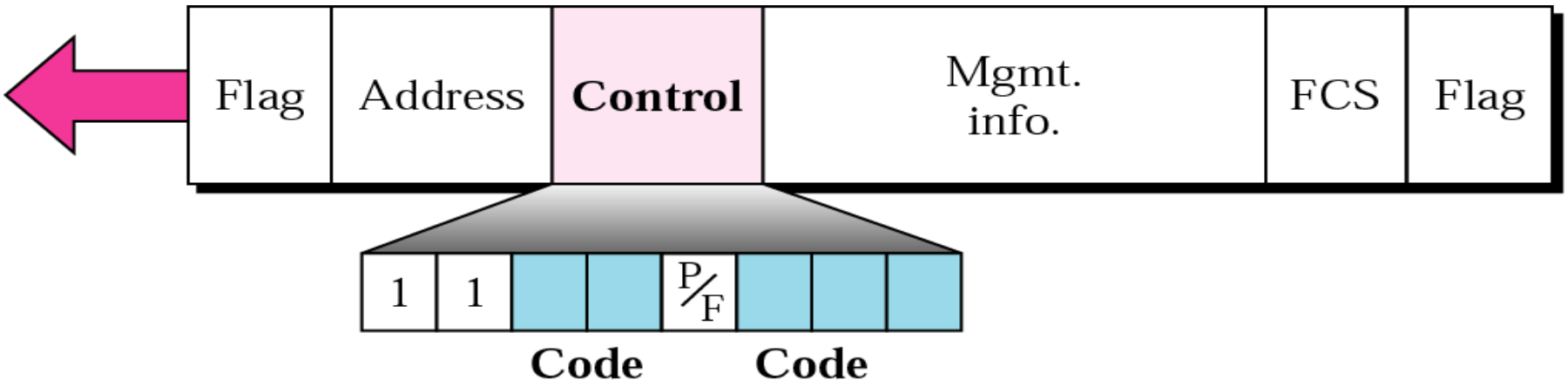
# S-frame control field in HDLC



# Control Field for U-Frames

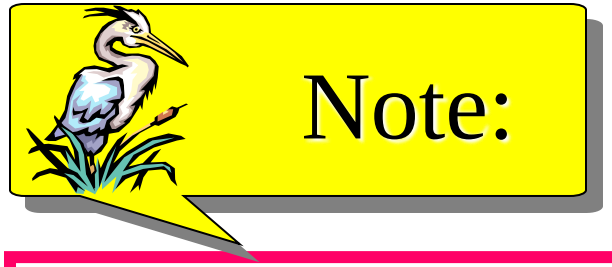
- Unnumbered frames
- to exchange session management and control information between connected devices
- contain an information field used for system management information
- Codes divided into two sections:
  - 2-bit prefix before P/F bit
  - 3-bit suffix after P/F bit
  - these 2 segments (5 bits) create up to 32 different types of U-frames

# U-frame control field in HDLC



# U-frame control command and response

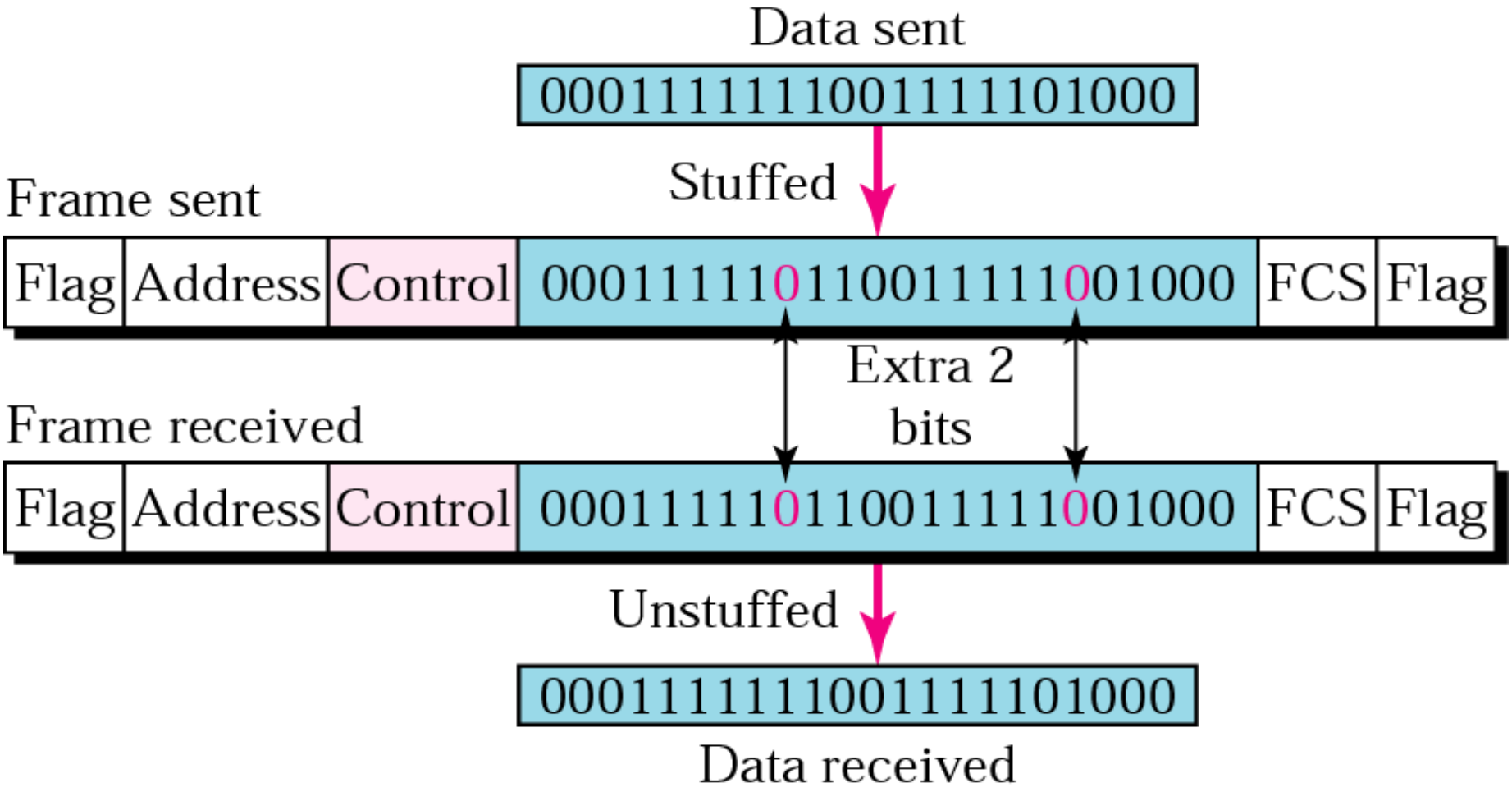
| *Code* | *Command* | *Response* | *Meaning* |
|---|---|---|---|
| **00 001** | SNRM | | Set normal response mode |
| **11 011** | SNRME | | Set normal response mode, extended |
| **11 100** | SABM | **DM** | Set asynchronous balanced mode or **disconnect mode** |
| **11 110** | SABME | | Set asynchronous balanced mode, extended |
| **00 000** | UI | **UI** | Unnumbered information |
| **00 110** | | **UA** | **Unnumbered acknowledgment** |
| **00 010** | DISC | **RD** | Disconnect or **request disconnect** |
| **10 000** | SIM | **RIM** | Set initialization mode or **request information mode** |
| **00 100** | UP | | Unnumbered poll |
| **11 001** | RSET | | Reset |
| **11 101** | XID | **XID** | Exchange ID |
| **10 001** | FRMR | **FRMR** | Frame reject |



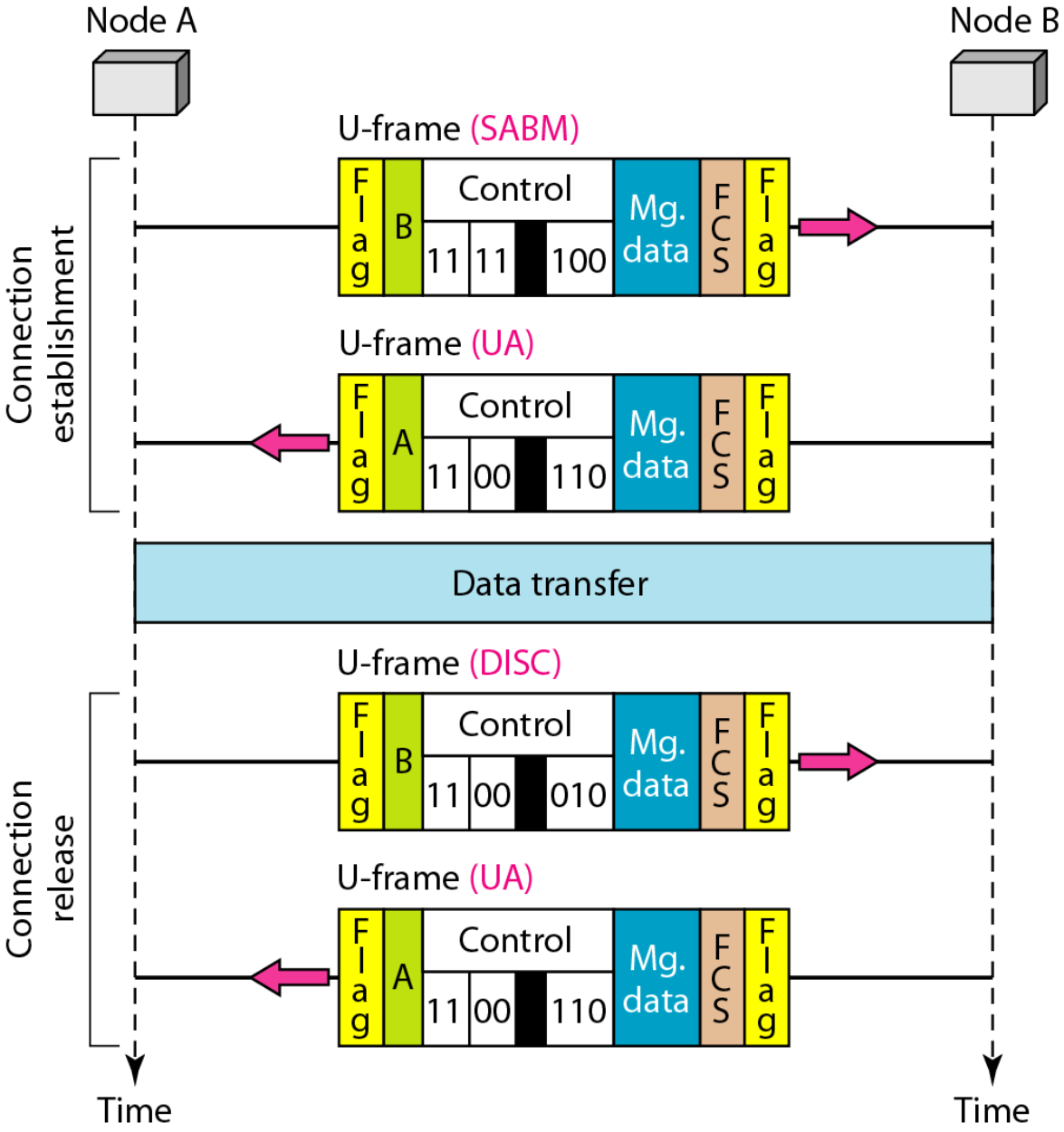
## Note:

*Bit stuffing is the process of adding one extra 0 whenever there are five consecutive 1s in the data so that the receiver does not mistake the data for a flag.*

# Bit stuffing and removal



# *Example of connection and disconnection using HDLC*




- ***Node A asks for connection with SABM frame***
- ***Node B gives positive response with an UA frame***
- ***After these data can be transferred between two***
- ***A sends a DISC frame to release the connection***
- ***Confirmed by B with UA***

# *Point-to-Point Access*

## *PPP*

# Point-to-Point Protocol (PPP)

- For point-to-point access
- PPP is a byte-oriented protocol
- Internet users need to connect their computers to server of an ISP with modem
- Telephone line provides services of physical layer
- To control and manage transfer of data, there is a need for PPP protocol at data link layer

# PPP provides several services:

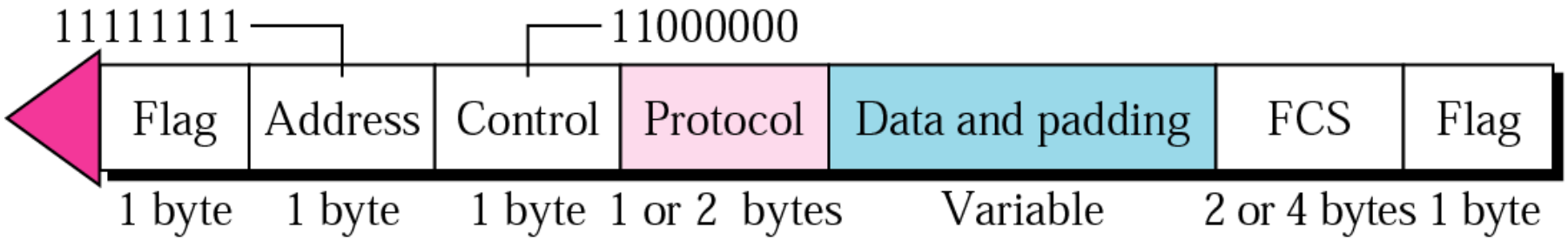
1. defines format of frame to be exchanged between devices
2. defines how 2 devices can negotiate establishment of link and exchange of data
3. defines how network layer data are encapsulated in data link frame
4. defines how 2 devices can authenticate each other
5. provides multiple network layer services supporting a variety of network layer protocols
6. provides connections over multiple links
7. provides network address configuration; useful when home user needs temporary network address to connect to Internet

# Several services are missing

1. does not provide flow control
2. has a very simple mechanism for error control
   - CRC field is used to detect errors
   - Corrupted frame is silently discarded
   - upper-layer protocol needs to take care of the problem
   - Lack of error control and sequence numbering may cause a packet to be received out of order
3. does not provide addressing mechanism to handle frames in a multipoint configuration

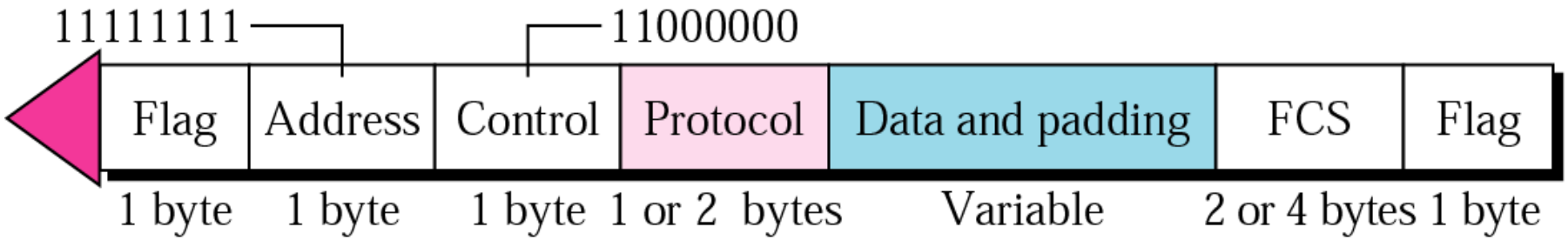
# Frame Format

- Flag-01111110-flag is treated as a byte not bit
- Address-constant value-11111111-broadcast address, 2 parties may agree to omit this byte
- Control-constant value-11000000- field is not needed, 2 parties can agree to omit this byte (no flow, error control)
- Protocol - defines what is carried in data field: user data or other information
  - default 2 bytes, 2 parties can agree to use 1 byte

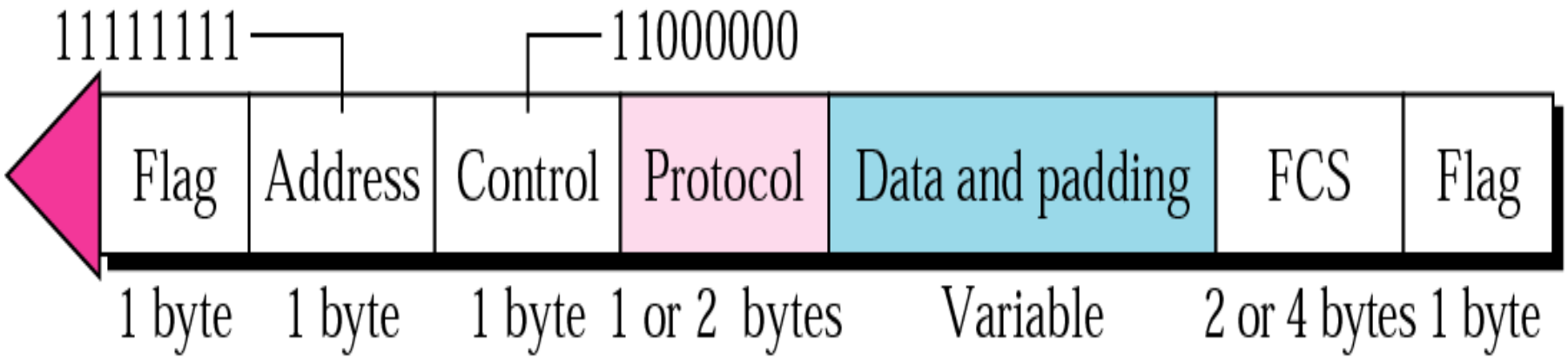


# Frame Format

- Payload field- user data/other information
  - data field = default of max. 1500 bytes; can be changed during negotiation
  - data field is byte stuffed if flag found in data
  - no field defining the size of the data field,
  - padding is needed if the size < max. default value or max. negotiated value
- FCS frame check sequence is 2/4 byte standard CRC



# PPP frame


11111111
11000000
Flag
Address
Control
Protocol
Data and padding
FCS
Flag
1 byte
1 byte
1 byte
1 or 2 bytes
Variable
2 or 4 bytes
1 byte

# Byte Stuffing

- Flag  is a byte and needs to be escaped whenever it appears in data section of frame
- PPP is a byte-oriented protocol using byte stuffing with the escape byte 01111101

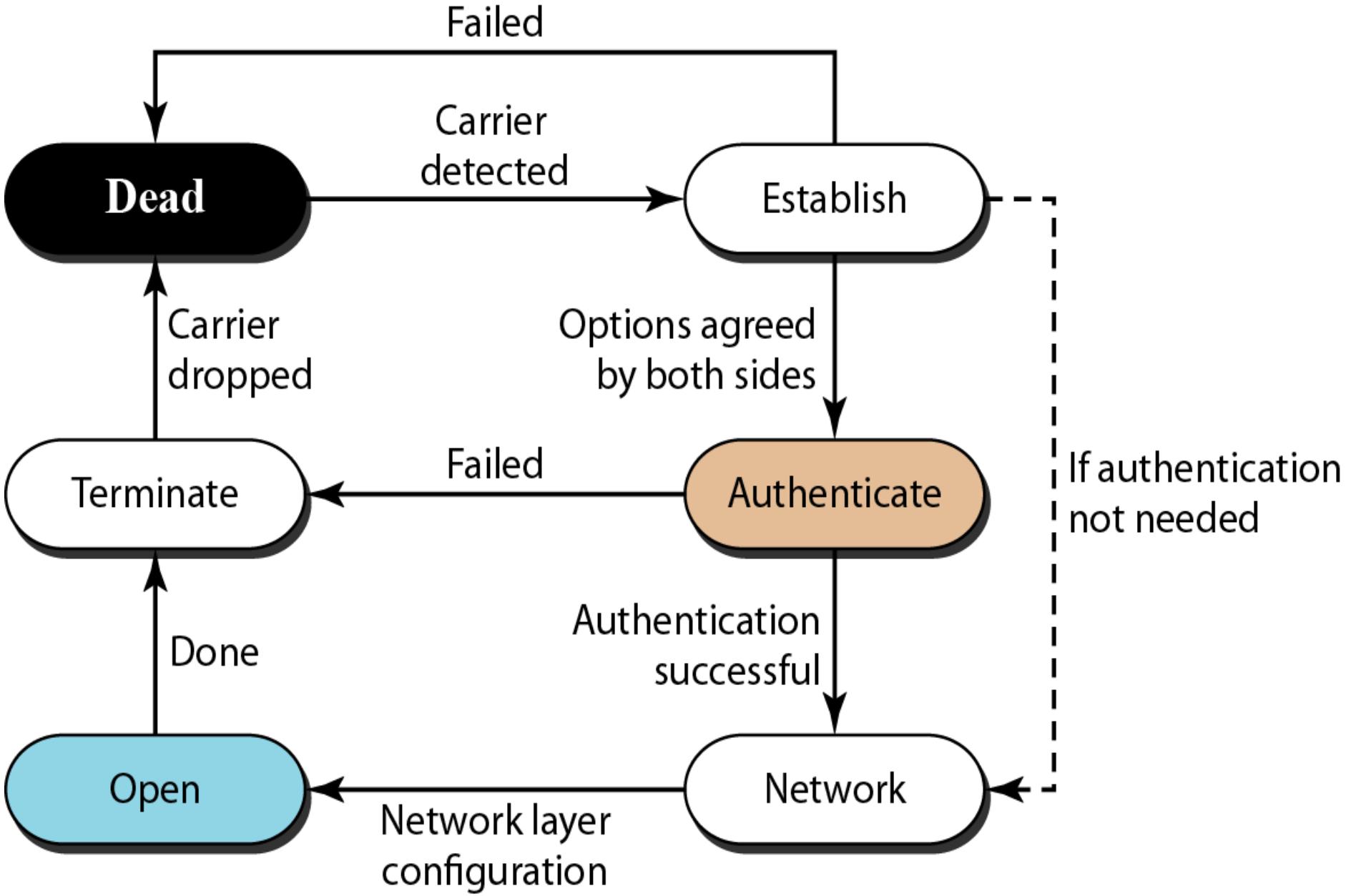
# Transition States of PPP connection

- Dead - link is not being used
- Establish- One node starts communication, options are negotiated, on success goes to authentication phase or to networking phase
- Authenticate- is optional, nodes decide during establishment phase
- Network-negotiation for network layer protocols , node is running multiple protocols simultaneously at network layer, receiving node needs to know which protocol will receive data

# Transition States of PPP connection

- Open - data transfer takes place until one node wants to terminate connection.
- Terminate - connection is terminated, packets are exchanged between 2 ends for house cleaning and closing link

# Transition States of PPP connection

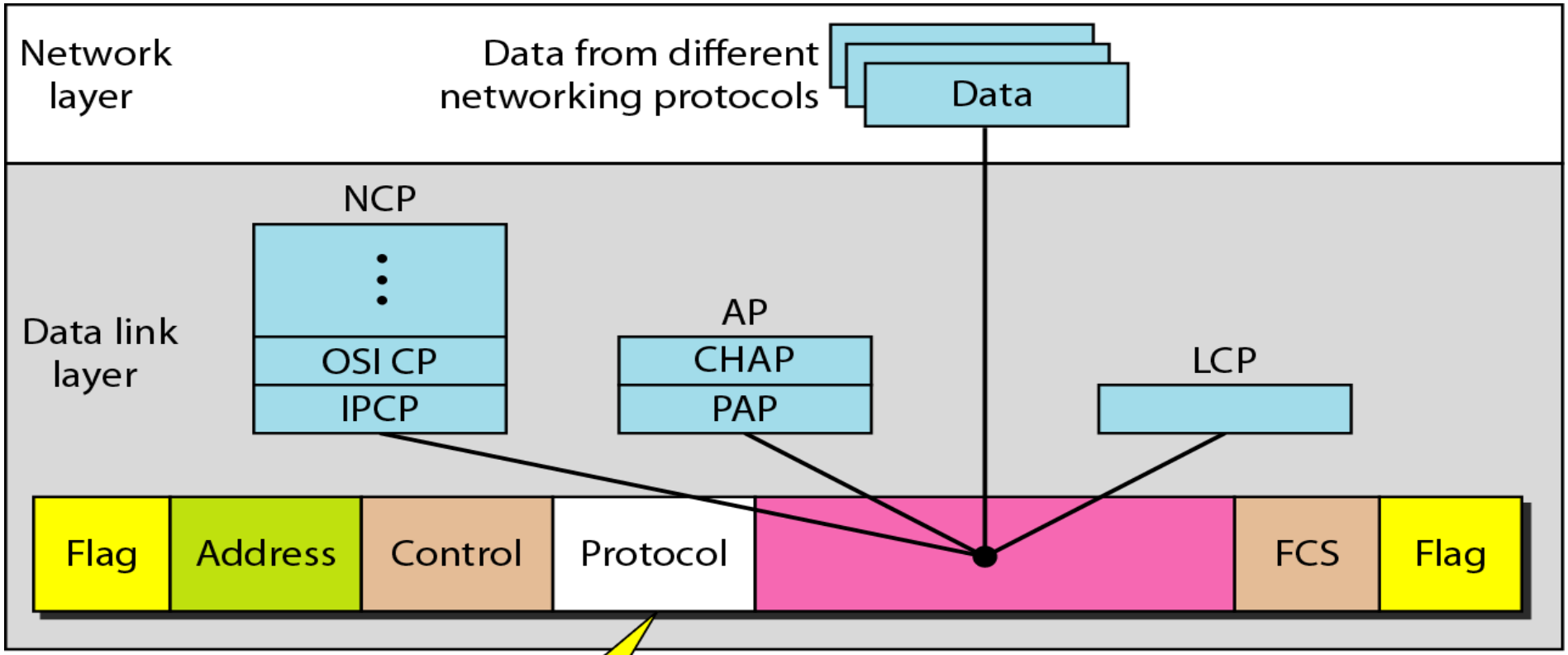


# Multiplexing

- PPP is a data link layer protocol
- uses another set of protocols to
  - establish the link,
  - authenticate the parties involved,
  - carry network layer data.
- Three  sets of protocols to make PPP powerful:
- Link Control Protocol (LCP),
- Two  Authentication Protocols (APs),
- Several Network Control Protocols (NCPs).

# Multiplexing in PPP

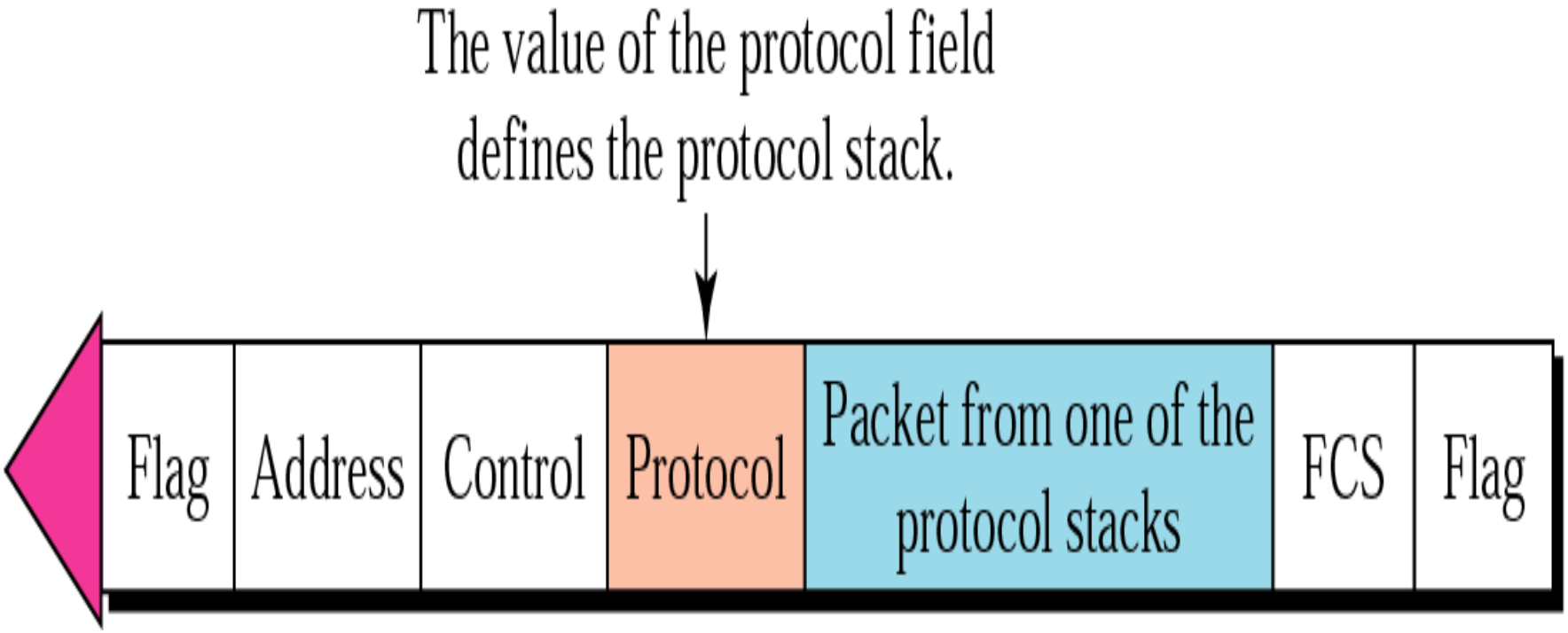
- PPP packet can carry data from one of the protocols in its data field
- Data may also come from several different network layers




LCP: 0xC021
AP: 0xC023 and 0xC223
NCP: 0x8021 and ....
Data: 0x0021 and ....

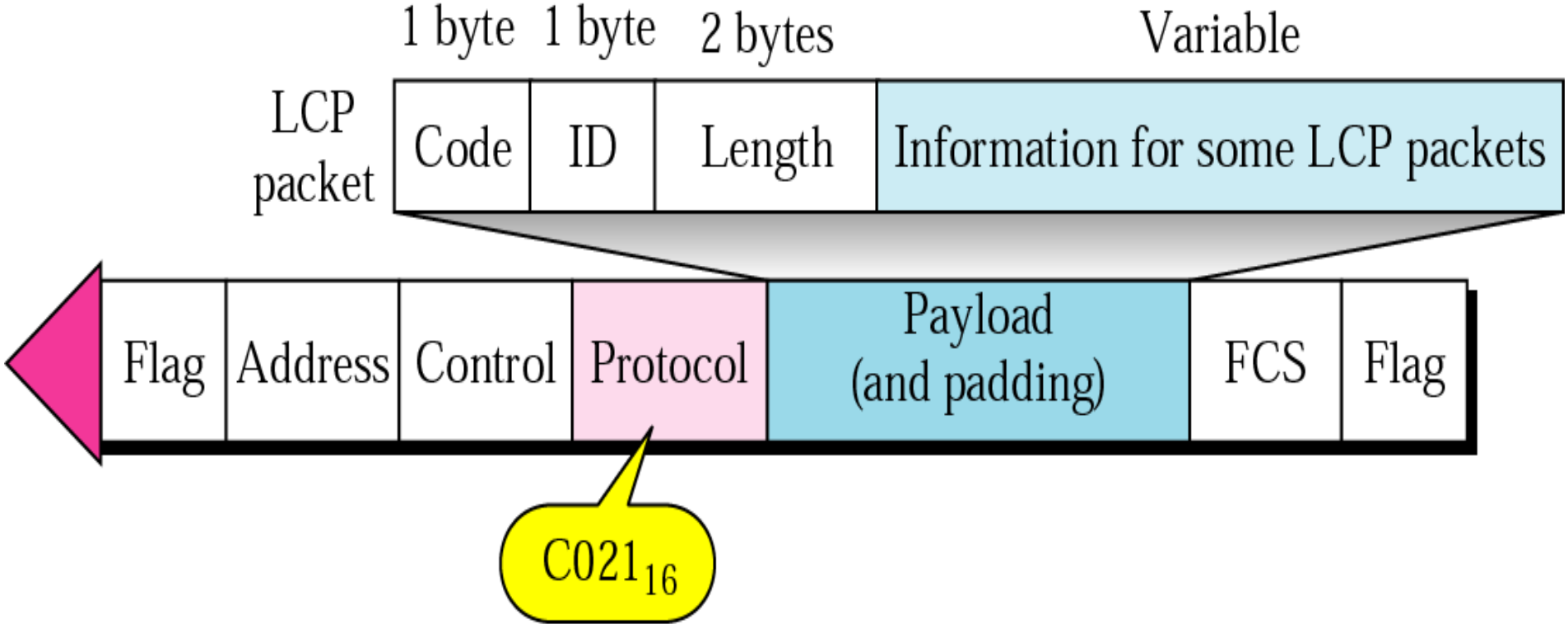
LCP: Link Control Protocol
AP: Authentication Protocol
NCP: Network Control Protocol

# Protocol stack



# Link Control Protocol

- for establishing, maintaining, configuring, terminating links
- LCP packets are carried in payload field of PPP frame with protocol field = 0xC021



# LCP packets and their codes

| *Code* | *Packet Type* | *Description* |
|---|---|---|
| 0x01 | Configure-request | Contains the list of proposed options and their values |
| 0x02 | Configure-ack | Accepts all options proposed |
| 0x03 | Configure-nak | Announces that some options are not acceptable |
| 0x04 | Configure-reject | Announces that some options are not recognized |
| 0x05 | Terminate-request | Request to shut down the line |
| 0x06 | Terminate-ack | Accept the shutdown request |
| 0x07 | Code-reject | Announces an unknown code |
| 0x08 | Protocol-reject | Announces an unknown protocol |
| 0x09 | Echo-request | A type of hello message to check if the other end is alive |
| 0x0A | Echo-reply | The response to the echo-request message |
| 0x0B | Discard-request | A request to discard the packet |

1-4 = link configuration during establish phase
5-6 = link termination during termination phase
7-11=link monitoring and debugging

# Common Options

| *Option* | *Default* |
|---|---|
| Maximum receive unit (payload field size) | 1500 |
| Authentication protocol | None |
| Protocol field compression | Off |
| Address and control field compression | Off |

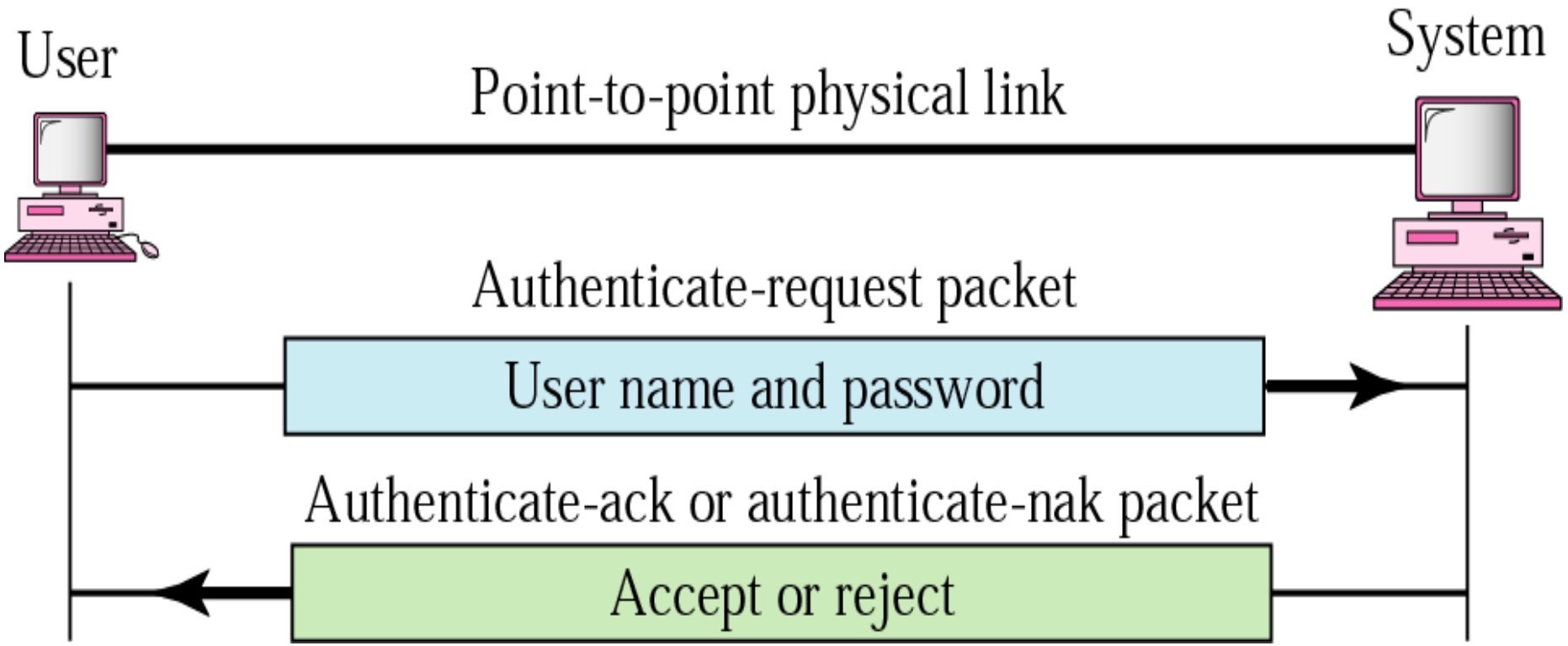
# Authentication Protocols

- Means validating the identity of a user who needs to access a set of resources (ISP)
  - Eg. user getting the services from ISP needs to be authenticated
- PPP has created 2 protocols
  - Password Authentication Protocol
  - Challenge Handshake Authentication Protocol.

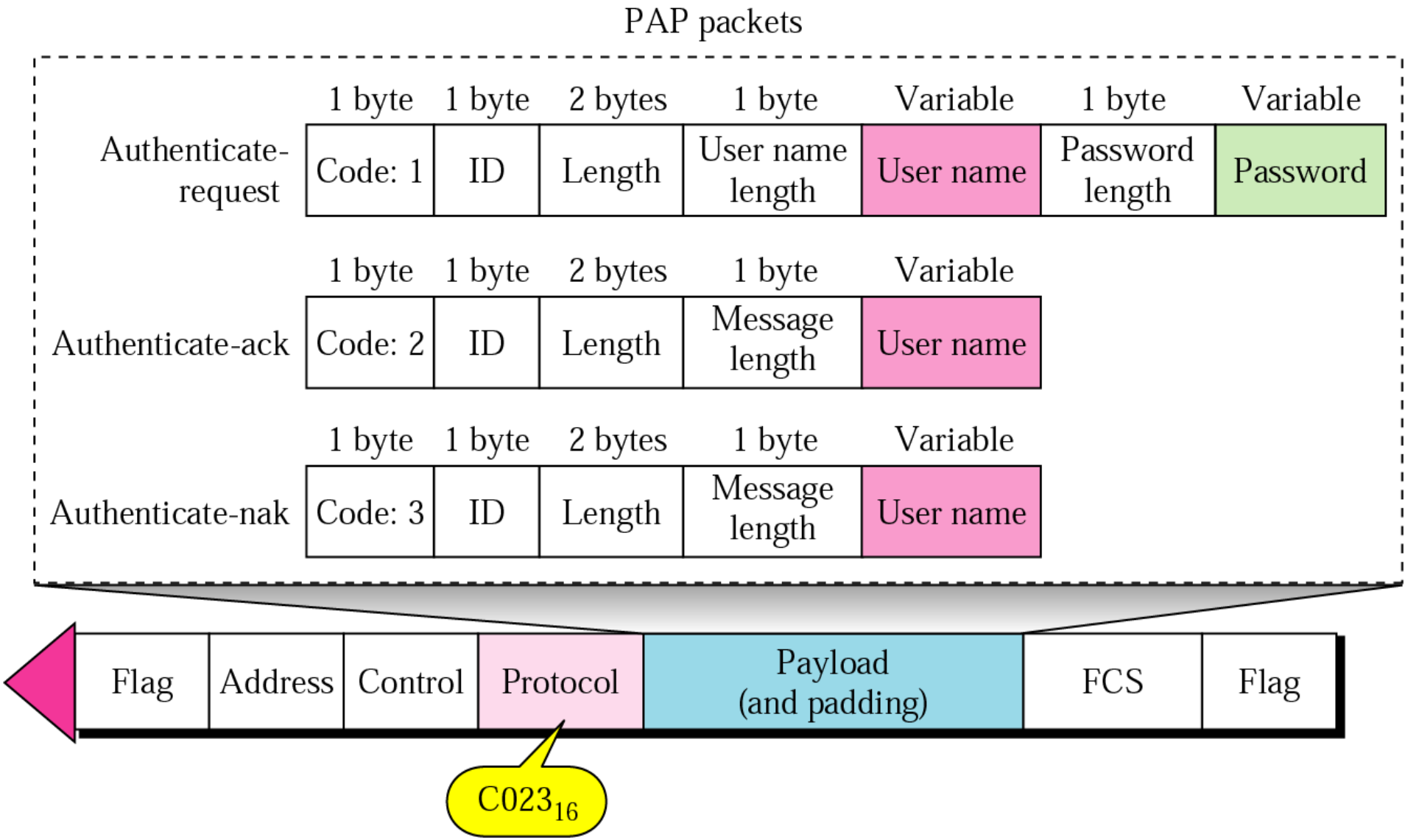
# Password Authentication Protocol (PAP)

- Two-step process:
  - user who wants to access a system sends an user name and password.
  - system checks the validity and either accepts or denies connection
- When PPP frame is carrying any PAP packets, value of protocol field = 0xC023
- Three PAP packets are authenticate-request, authenticate-ack, and authenticate-nak.

# PAP


User
System
Point-to-point physical link
Authenticate-request packet
User name and password
Authenticate-ack or authenticate-nak packet
Accept or reject

# PAP packets encapsulated in a PPP



# Challenge Handshake Authentication Protocol (CHAP)

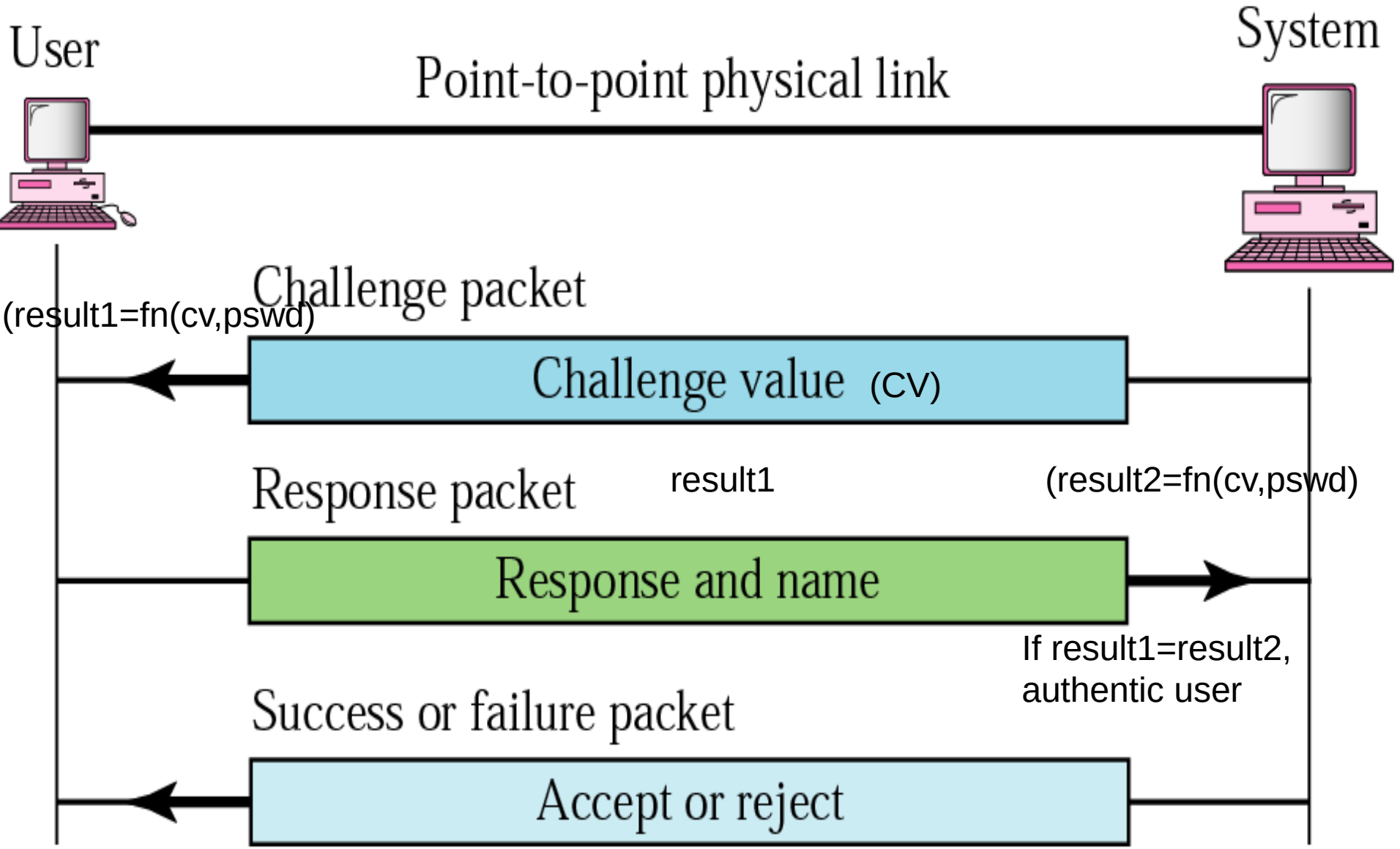
- 3 way hand-shaking authentication protocol
- provides greater security than PAP
- password is kept secret; it is never sent online.

# Challenge Handshake Authentication Protocol (CHAP)

- system sends user a challenge packet with a challenge value, usually a few bytes
- user applies a predefined function that takes challenge value, user's own password and creates a result, sends the result in response packet to system (result=fn(cv,pswd)
- System applies the same function to password of user and challenge value to create a result. (result=fn(cv, pswd)
- If the result created is the same as the result sent in the response packet, access is granted; otherwise, denied (result match=authentic user)
- the system continuously changes the challenge value.
- if the intruder learns the challenge value and the result, the password is still secret

# CHAP

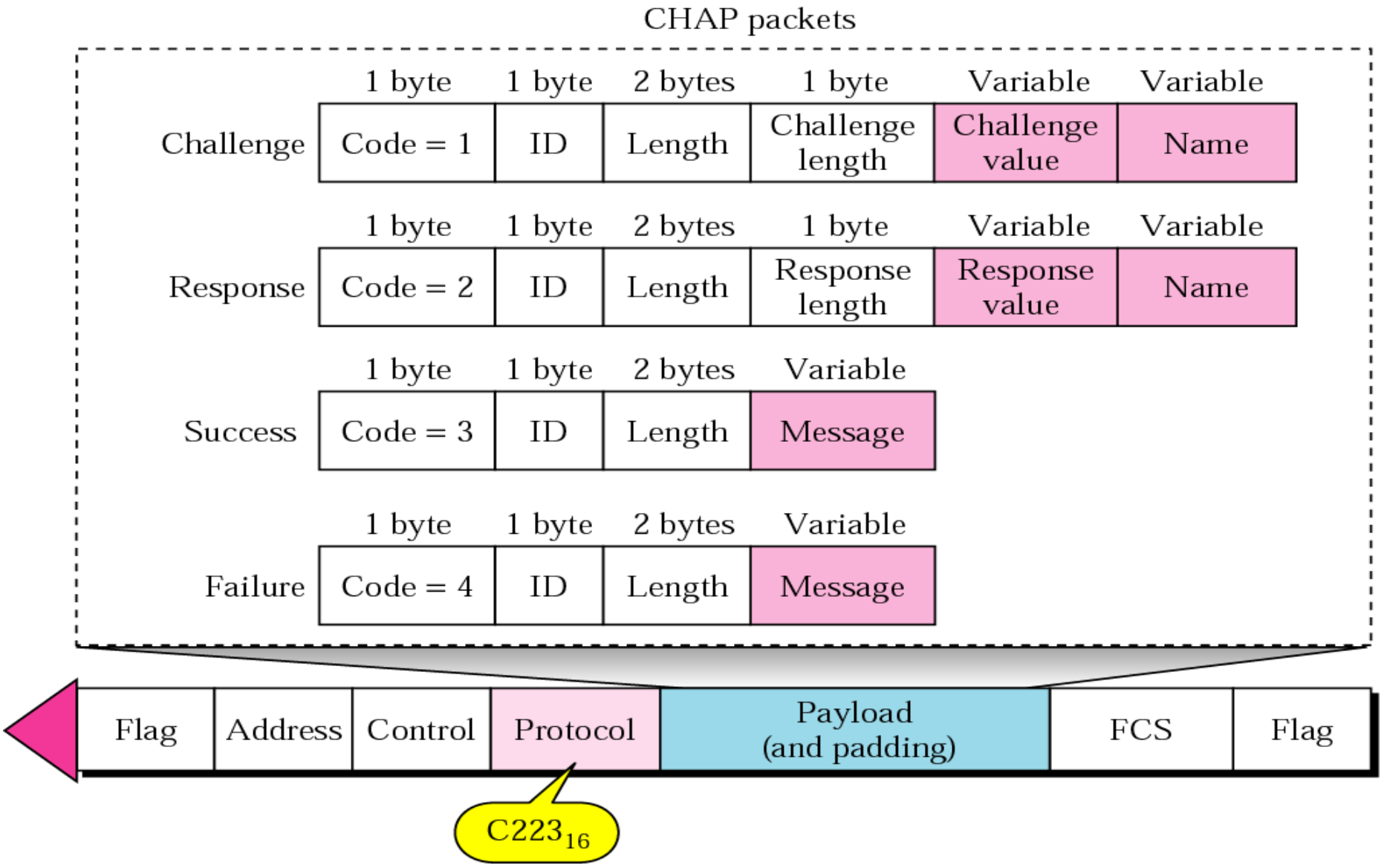
- system continuously changes cv
- if intruder can know cv and result but password is still secret



# Four CHAP packets

- challenge, response, success, and failure.
  - to send the challenge value
  - to return the result of the calculation
  - to allow access to the system
  - to deny access to the system.

# Four CHAP packets

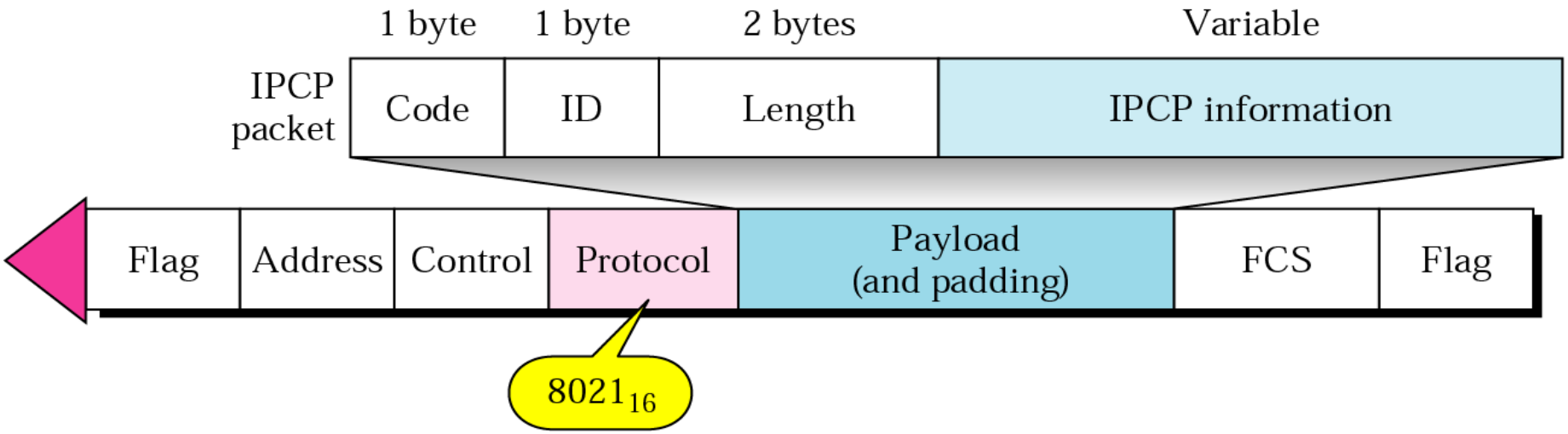


# Network Control Protocols

- PPP is a multiple-network layer protocol.
- can carry a network layer data packet from - Internet, OSI, Xerox, DECnet, AppleTalk, Novel …
- PPP has defined a specific Network Control Protocol for each network protocol.
- Eg. IPCP (Internet Protocol Control Protocol) configures link for carrying IP data packets.
- Xerox CP for the Xerox protocol data packets,
- none of the NCP packets carry network layer data;
- they configure link at network layer for incoming data.

# IPCP

- Configures link used to carry IP packets in Internet
- IPCP packet encapsulated in PPP frame



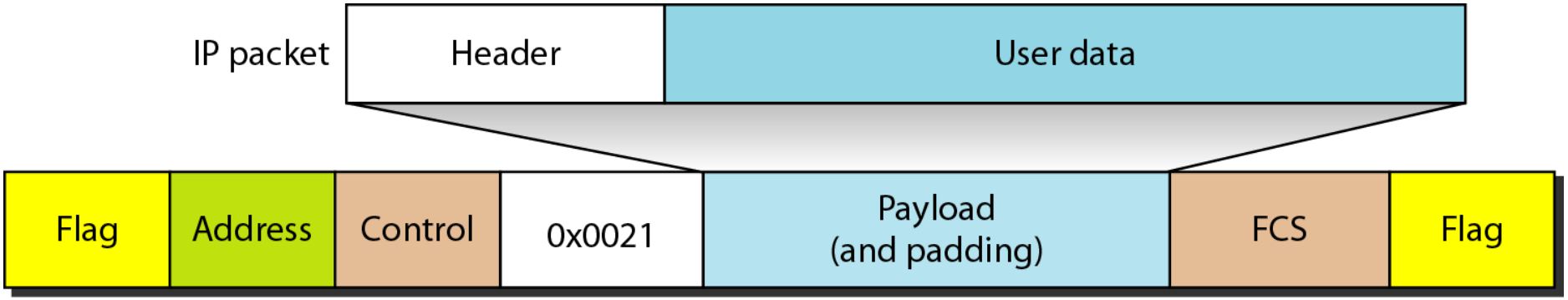
IPCP defines 7 packets, distinguished by their code values, code value for IPCP packets

| *Code* | *IPCP Packet* |
|---|---|
| 0x01 | Configure-request |
| 0x02 | Configure-ack |
| 0x03 | Configure-nak |
| 0x04 | Configure-reject |
| 0x05 | Terminate-request |
| 0x06 | Terminate-ack |
| 0x07 | Code-reject |

# Data from the Network Layer

- After network layer configuration is completed by one of the NCP protocols, users can exchange data packets from network layer
- there are different protocol fields for different network layers
- Eg. if PPP is carrying data from IP network layer, field value is 0021
- If PPP is carrying data from the OSI network layer, protocol field is 0023

# IP datagram encapsulated in a PPP frame



Thank You!