Data Link Control (DLC): HDLC & PPP

Dr Samayveer Singh

Dept. of Computer Science and Engineering

National Institute of Technology

Jalandhar, Punjab, India

samays@nitj.ac.in

High-level Data Link Control (HDLC)

- ***HDLC is a bit -oriented protocol for communication over point-to-point and multipoint links.***
- ***It implements the Stop-and-Wait protocol we discussed earlier.***
- ***Although this protocol is more a theoretical issue than practical, most of the concept defined in this protocol is the basis for other practical protocols such as PPP, Ethernet, or wireless LANs.***

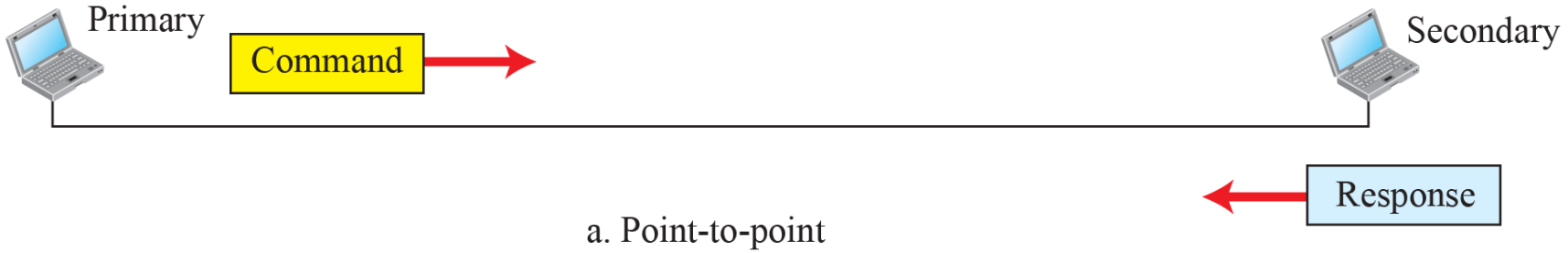
1 Transfer Modes

HDLC provides two common transfer modes that can be used in different configurations:

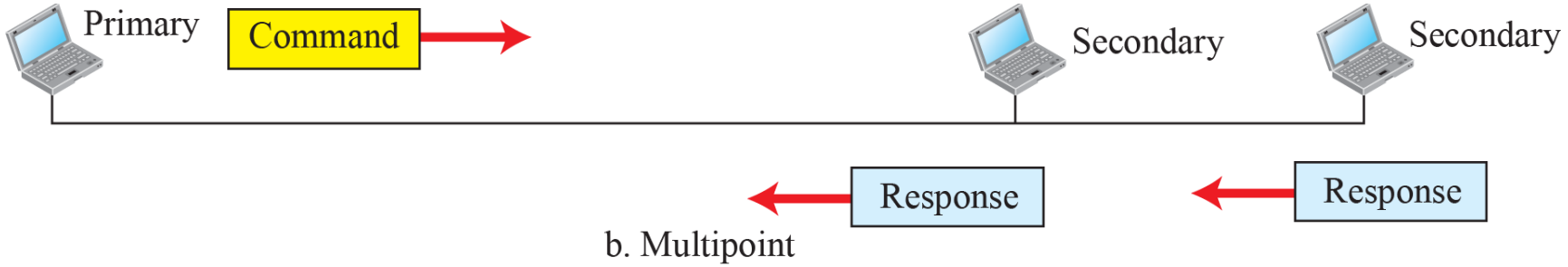
Normal response mode (NRM) and

Asynchronous balanced mode (ABM)

Figure 1: Normal response mode



a. Point-to-point



b. Multipoint

Figure 2: Asynchronous balanced mode

2 Framing

To provide the flexibility necessary to support all the options possible in the modes and configurations just described, HDLC defines three types of frames:

Information frames (I-frames),

Supervisory frames (S-frames), and

Unnumbered frames (U-frames).

Figure 3: HDLC frames

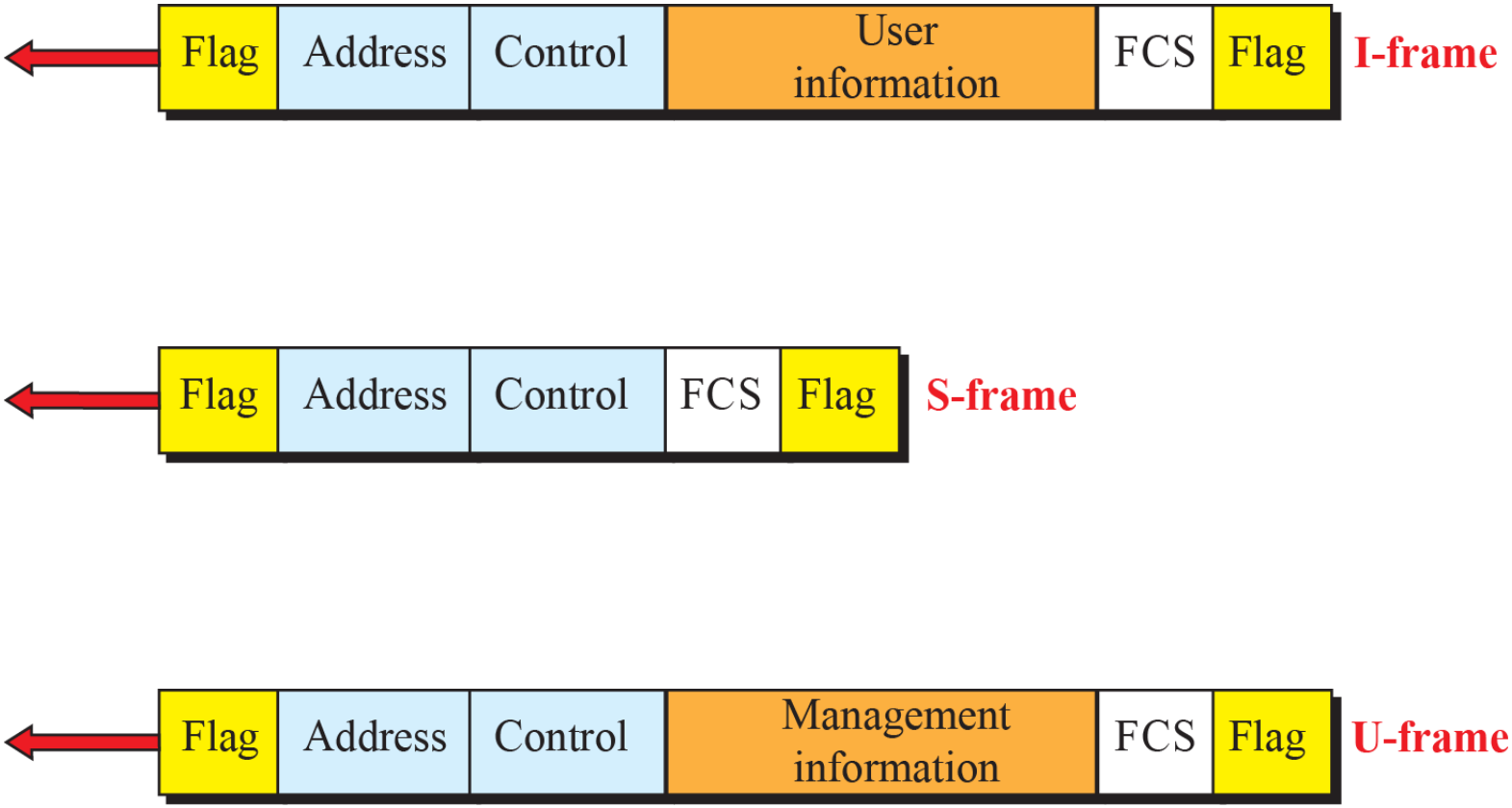
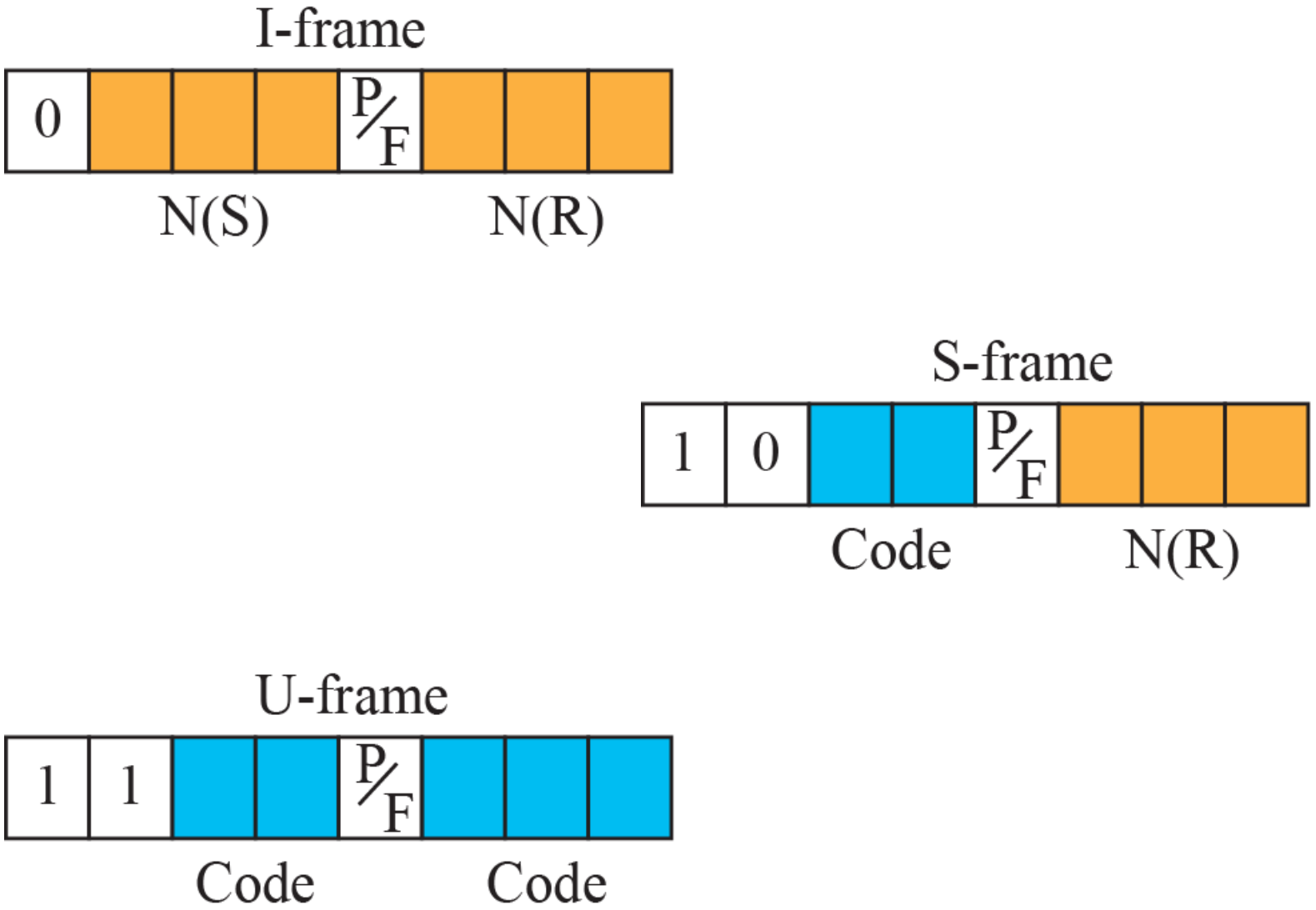


Figure 4: ***Control field format for the different frame types***

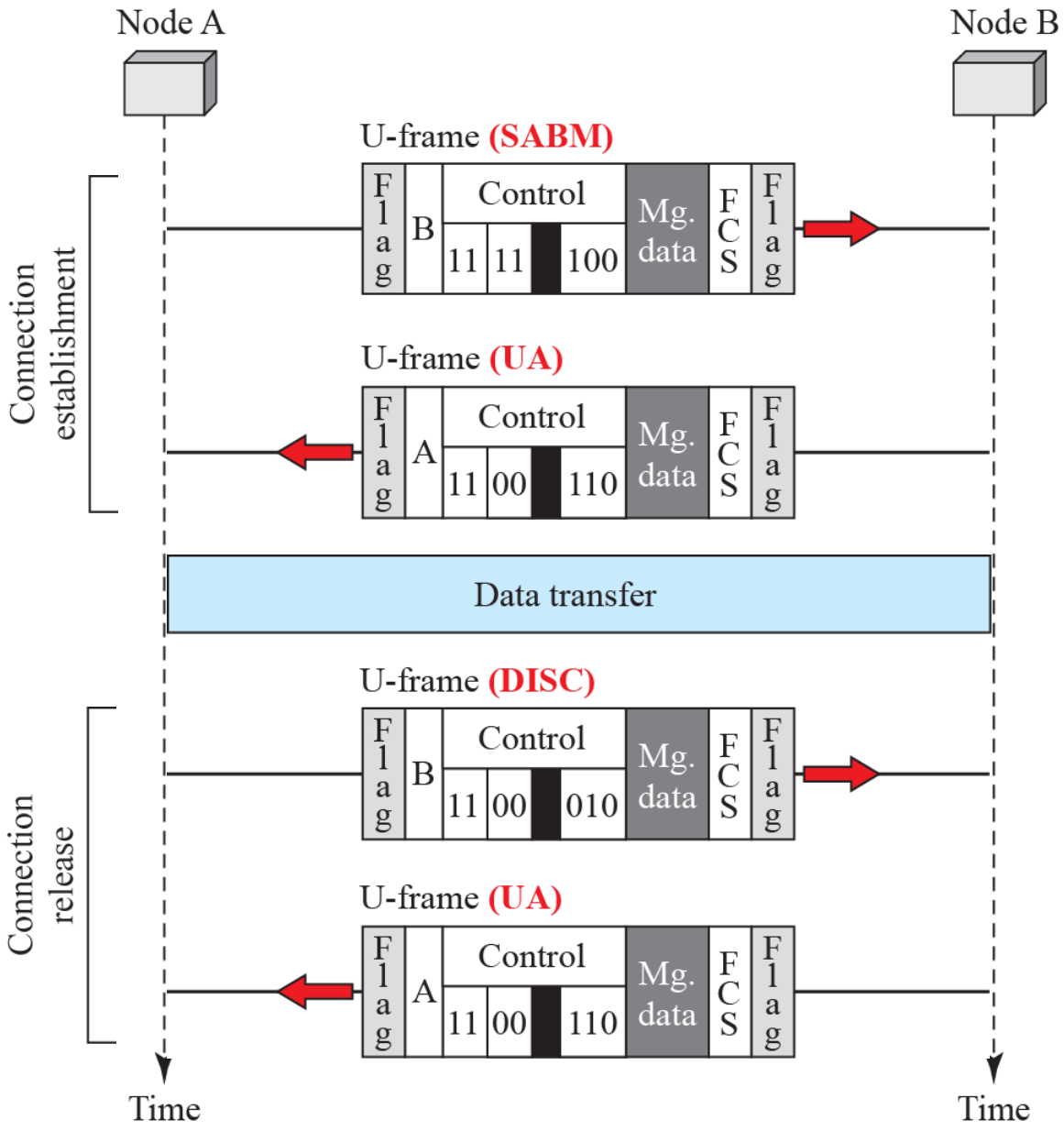


Example 1

Figure 5 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 5: Example of connection and disconnection



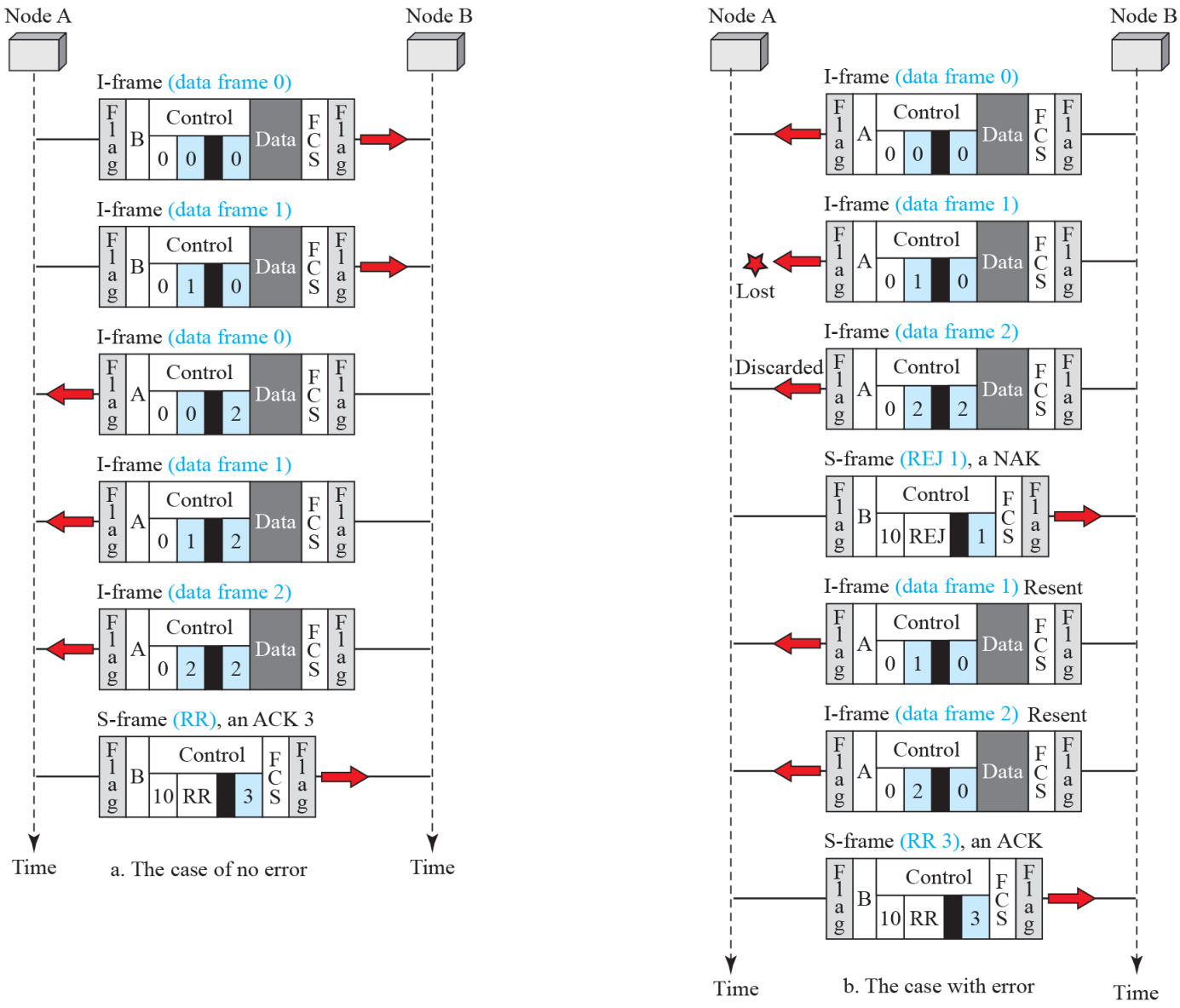
Example 2

Figure 6 shows two exchanges using piggybacking.

The first is the case where no error has occurred;

the second is the case where an error has occurred and some frames are discarded.

Figure 6: Example of piggybacking with and without error



PPP

- ***One of the most common protocols for point-to-point access is the Point-to-Point Protocol (PPP).***
- ***Today, millions of Internet users who need to connect their home computers to the server of an Internet service provider use PPP.***
- ***To control and manage the transfer of data, there is a need for a point-to-point protocol at the data-link layer.***
- ***PPP is by far the most common.***

1 Services

The designers of PPP have included several services to make it suitable for a point-to-point protocol, but have ignored some traditional services to make it simple.

2 Framing

PPP uses a character-oriented (or byte-oriented) frame. Figure 7 shows the format of a PPP frame.

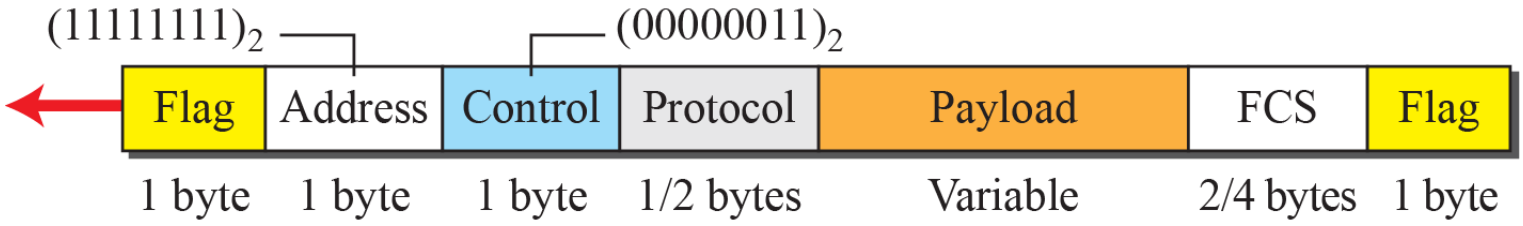


Figure 7: PPP frame format

3 Transition Phases

A PPP connection goes through phases which can be shown in a transition phase diagram.

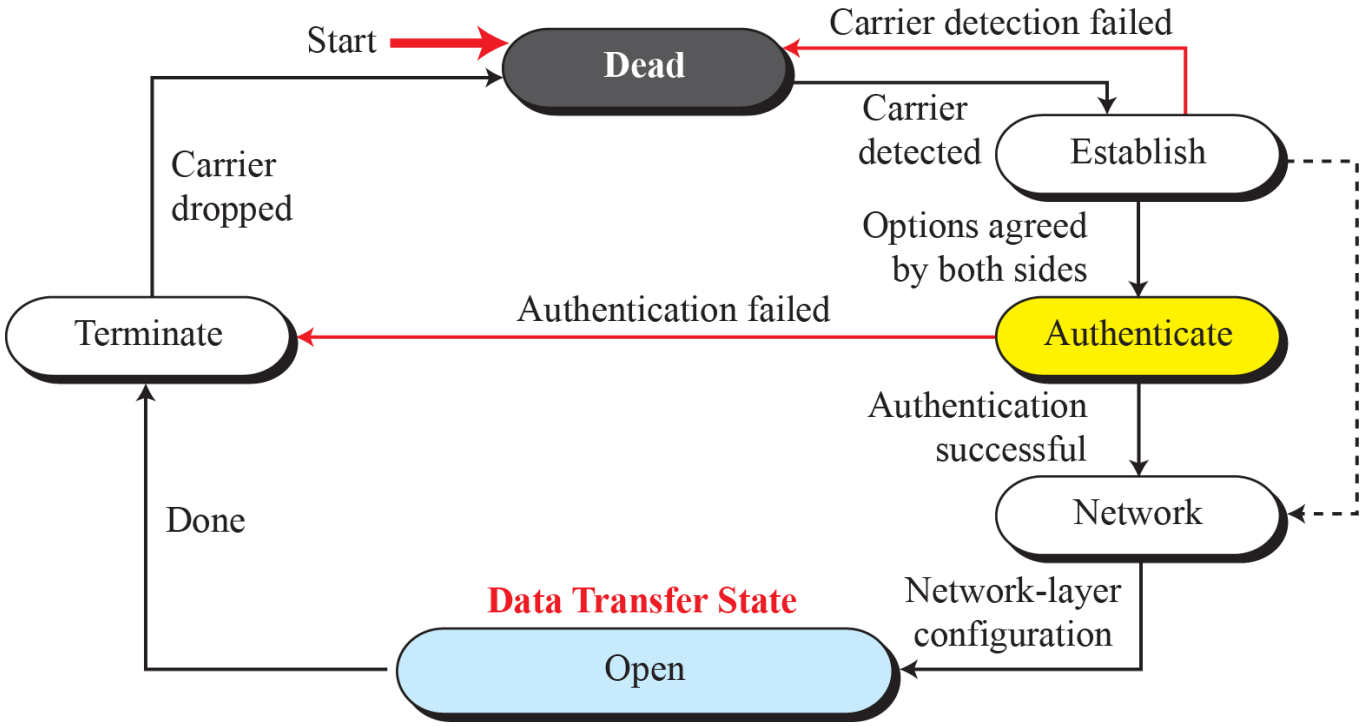


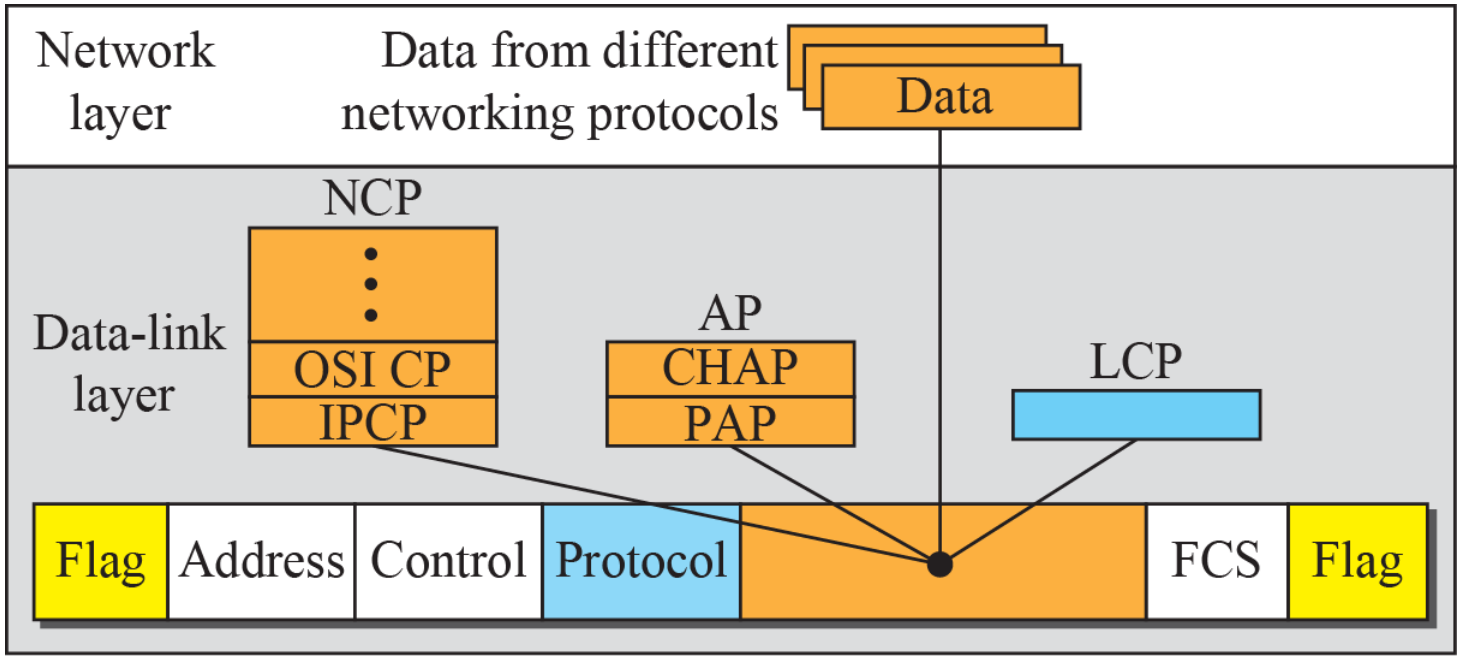
Figure 8: Transition phases

4 Multiplexing

Although PPP is a link-layer protocol, it uses another set of protocols to establish the link, authenticate the parties involved, and carry the network-layer data.

Three sets of protocols are defined to make PPP powerful: the Link Control Protocol (LCP), two Authentication Protocols (APs), and several Network Control Protocols (NCPs).

Figure 9: Multiplexing in PPP



Legend

LCP : Link control protocol
AP : Authentication protocol
NCP: Network control protocol

Protocol values:

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

Figure 10: LCP packet encapsulated in a frame

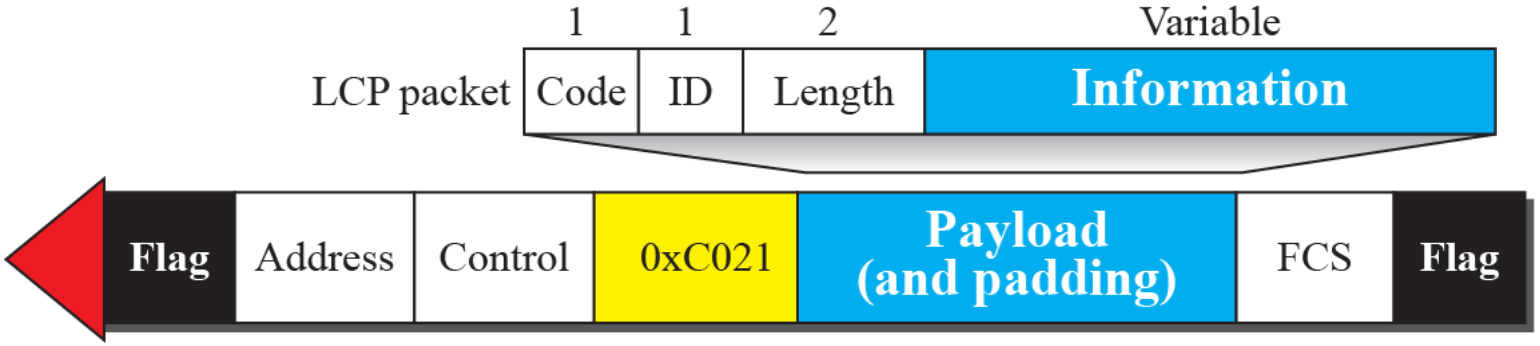


Table 1: LCP Packets

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

Table 2: Common options

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

Figure 11: PAP packets encapsulated in a PPP frame

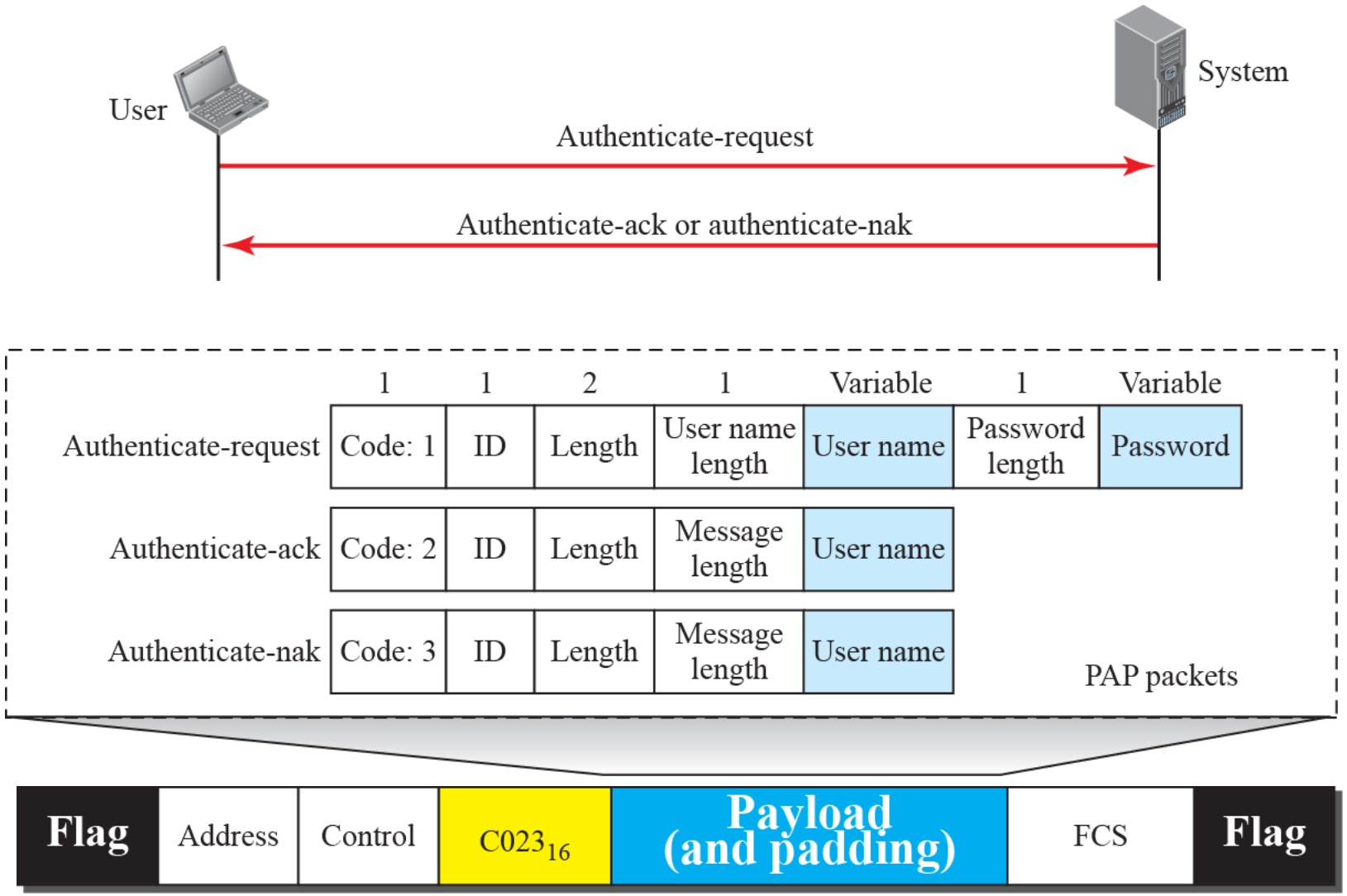
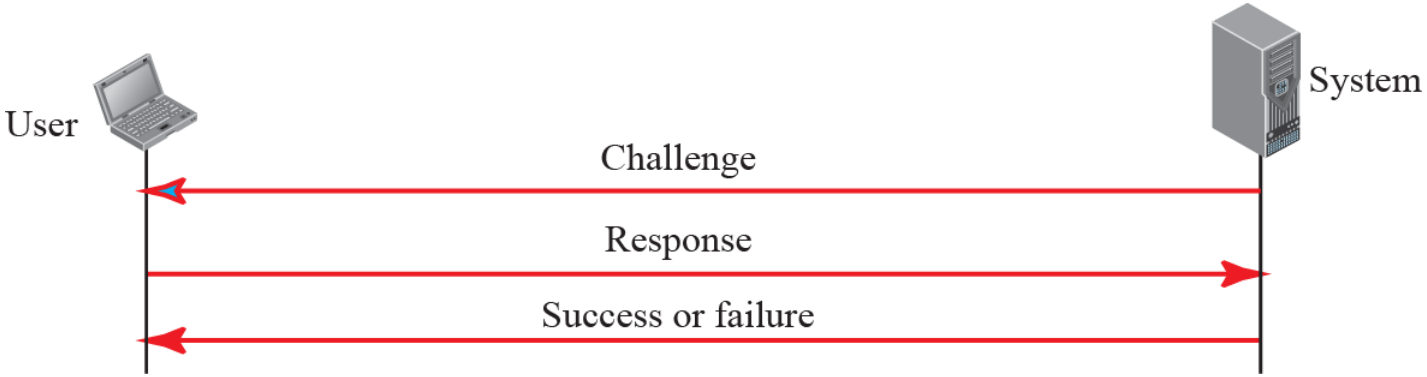


Figure 12: CHAP packets encapsulated in a PPP frame



	1	1	2	1	Variable	Variable
Challenge	Code: 1	ID	Length	Challenge length	Challenge value	Name
Response	Code: 2	ID	Length	Response length	Response value	Name
Success	Code: 3	ID	Length	Message		
Failure	Code: 4	ID	Length	Message		

CHAP packets

Flag	Address	Control	0xC223	Payload (and padding)	FCS	Flag

Figure 13: IPCP packet encapsulated in PPP frame

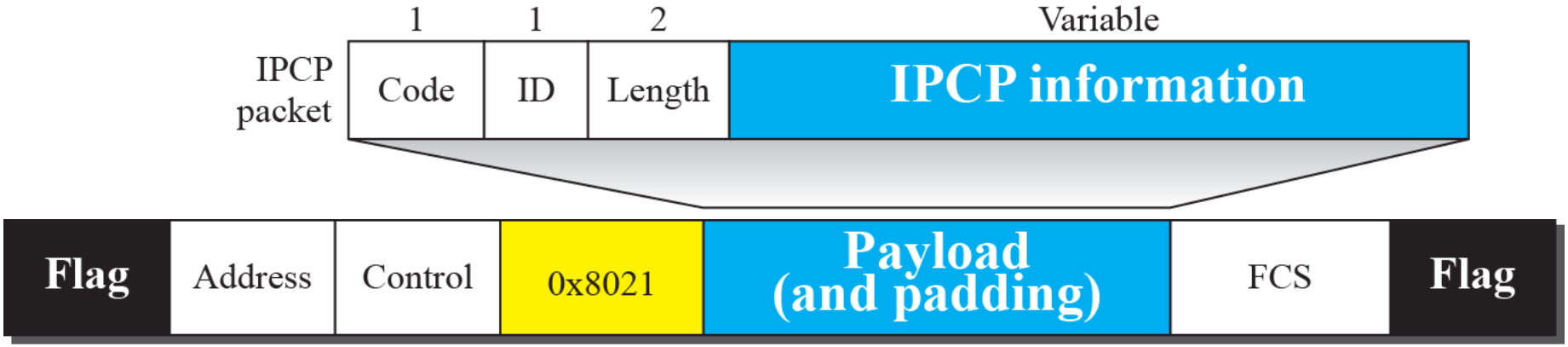


Table 3: Code values for IPCP Packets

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

Figure 14: IP datagram encapsulated in a PPP frame

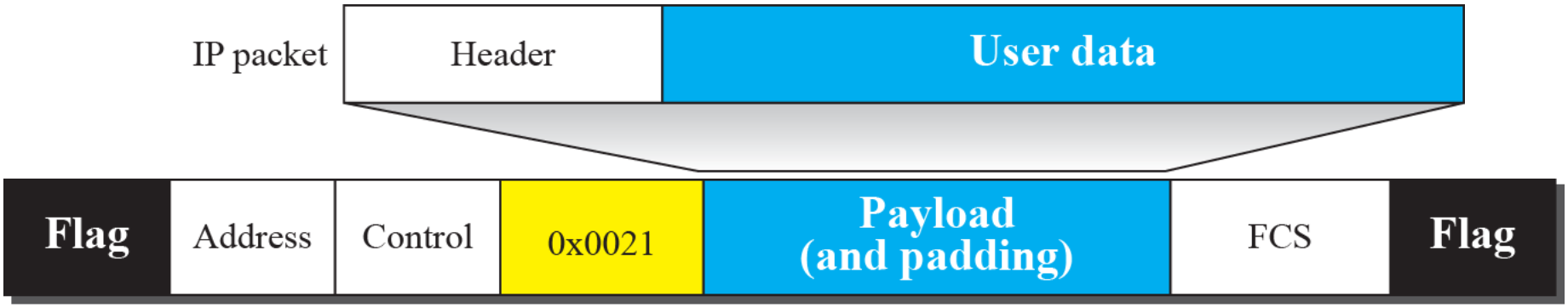
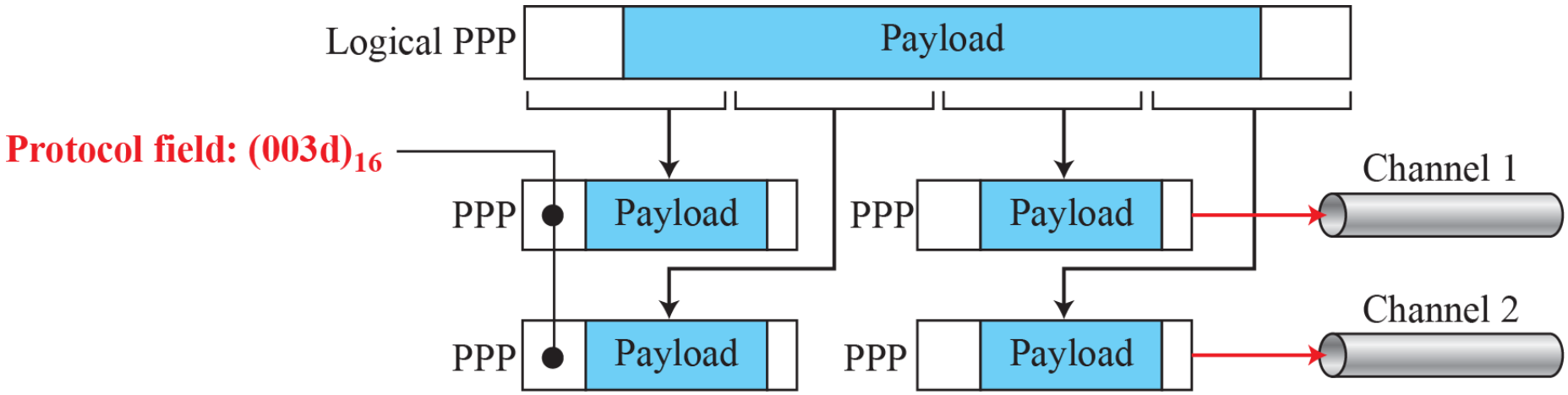


Figure 15: Multilink PPP



Example 3

Let us go through the phases followed by a network layer packet as it is transmitted through a PPP connection.

Figure 16 shows the steps.

For simplicity, we assume unidirectional movement of data from the user site to the system site (such as sending an e-mail through an ISP).

Figure 16: An example