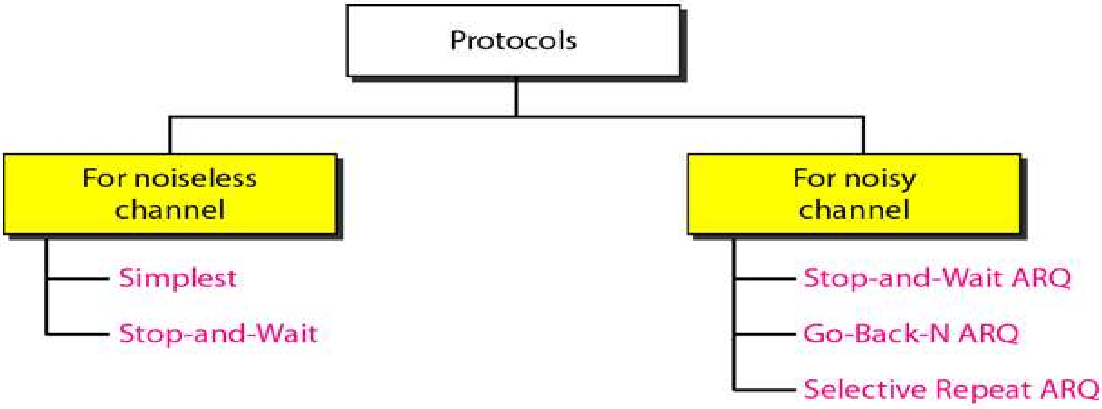
UNIT-3 (Part-2)
DATA LINK LAYER Protocols

Syllabus:
Elementary Data Link Layer protocols: simplex protocol, Simplex stop and wait, Simplex protocol for Noisy Channel. Sliding window protocol: One bit, Go back N, Selective repeat-Stop and wait protocol, Data link layer in HDLC: configuration and transfer modes, frames, control field, point to point protocol (PPP): framing transition phase, multiplexing, multi-link PPP.

Prtocols:

- A **protocol** is a set of rules and guidelines for communicating data. Rules are defined for each step and process during communication between two or more computers. Networks have to follow these rules to successfully transmit data.
- The data link layer can combine framing, flow control, and error control to achieve the delivery of data from one node to another.
- The protocols are normally implemented in software by using one of the common programming languages.



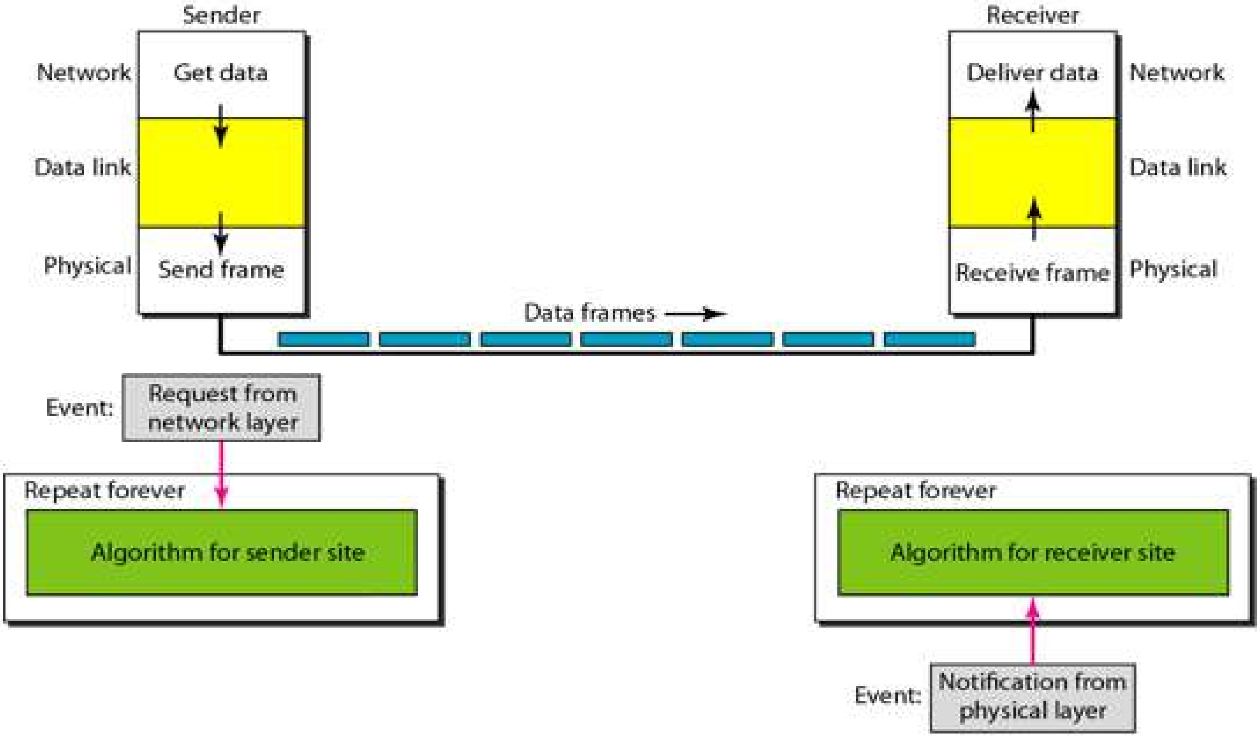
- **Noise-less channel (Error-free):**
  If an ideal channel, then no frames are lost, duplicated, or corrupted. For noise-less channel we have two protocols
    - Simplest Protocol
    - Stop-and-Wait Protocol

Simplest Protocol:

- Simplest protocol has no flow control and control. It is also called as unidirectional protocol in which data frames are travelling in only one direction from the sender to receiver.
- We assume that receiver immediately handle any frames. The datalink layer immediately removes header from the frame and handover to network layer, which can also accept packets immediately.

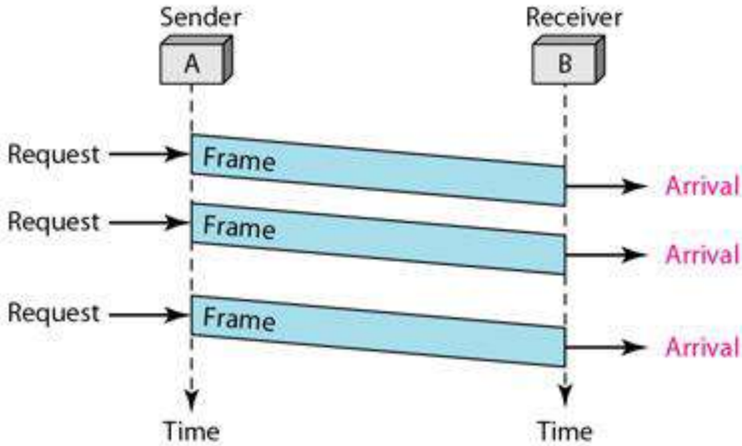
Design

There is no need for flow control in this scheme. The data link layer at the sender site gets data from its network layer, makes a frame out of the data, and sends it. The data link layer at the receiver site receives a frame from its physical layer, extracts data from the frame, and delivers the data to its network layer. The data link layers of the sender and receiver provide transmission services for their network layers. The data link layers use the services provided by their physical layers (such as signaling, multiplexing, and so on) for the physical transmission of bits.



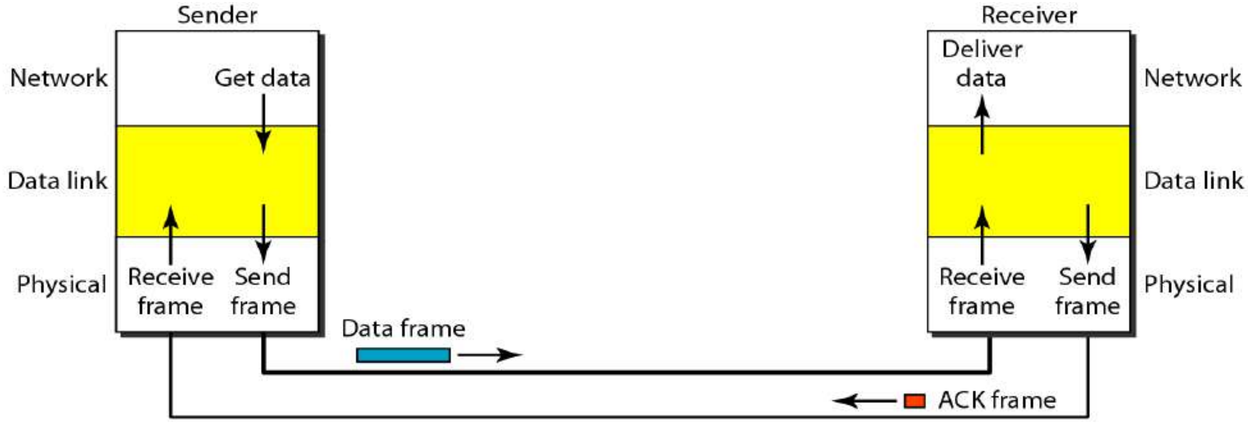
Flow diagram:

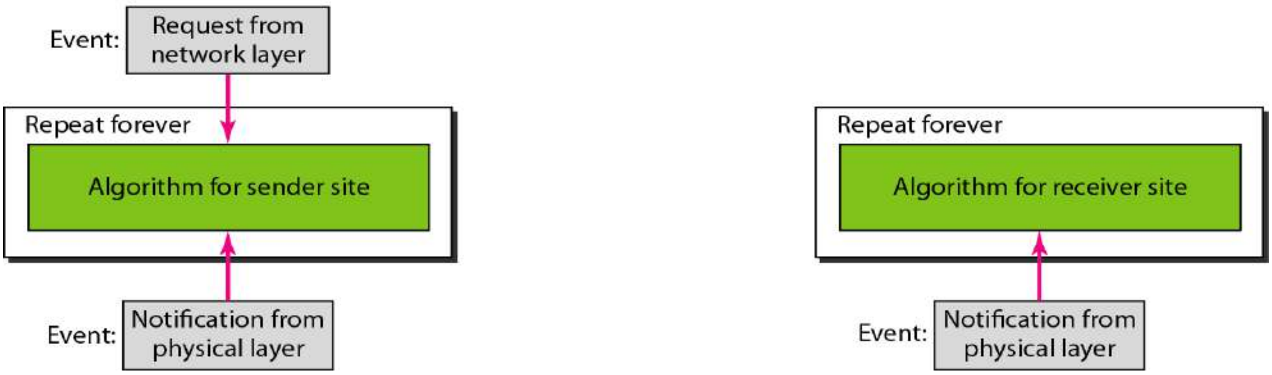
- Bellow figure shows an example of communication using this protocol. It is very simple.
- The sender sends a sequence of frames without even thinking about the receiver. To send three frames, three events occur at the sender site and three events at the receiver site.
- Note that the data frames are shown by tilted boxes; the height of the box defines the transmission time difference between the first bit and the last bit in the frame.



Stop-and-wait protocol:

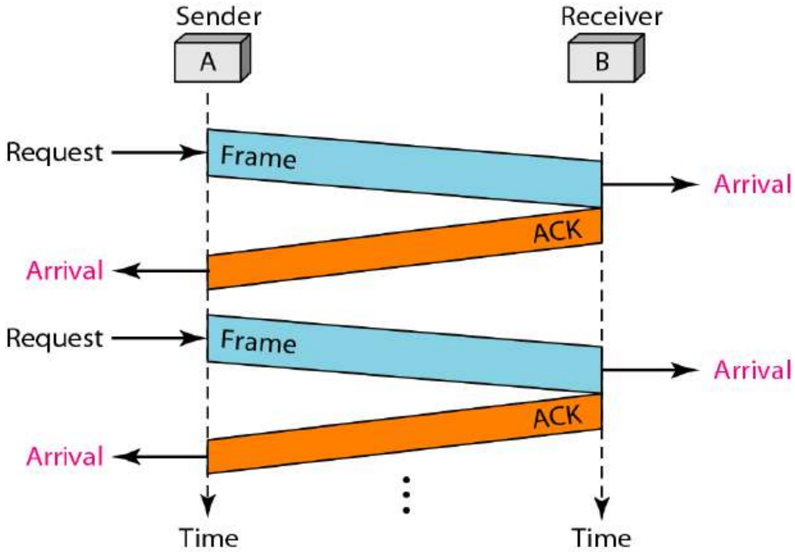
- In stop-and-wait protocol
  sender sends one frame, stops until it receives confirmation from the receiver (okay to go ahead), and then sends the next frame. We still have unidirectional communication for data frames, but auxiliary ACK frames (simple tokens of acknowledgment) travel from the other direction. We add flow control to our Simplest protocol.
- **Design of stop-and-wait protocol**





Flow diagram:

Bellow figure shows an example of communication using this protocol. It is still very simple. The sender sends one frame and waits for feedback from the receiver. When the ACK arrives, the sender sends the next frame. Note that sending two frames in the protocol involves the sender in four events and the receiver in two events.



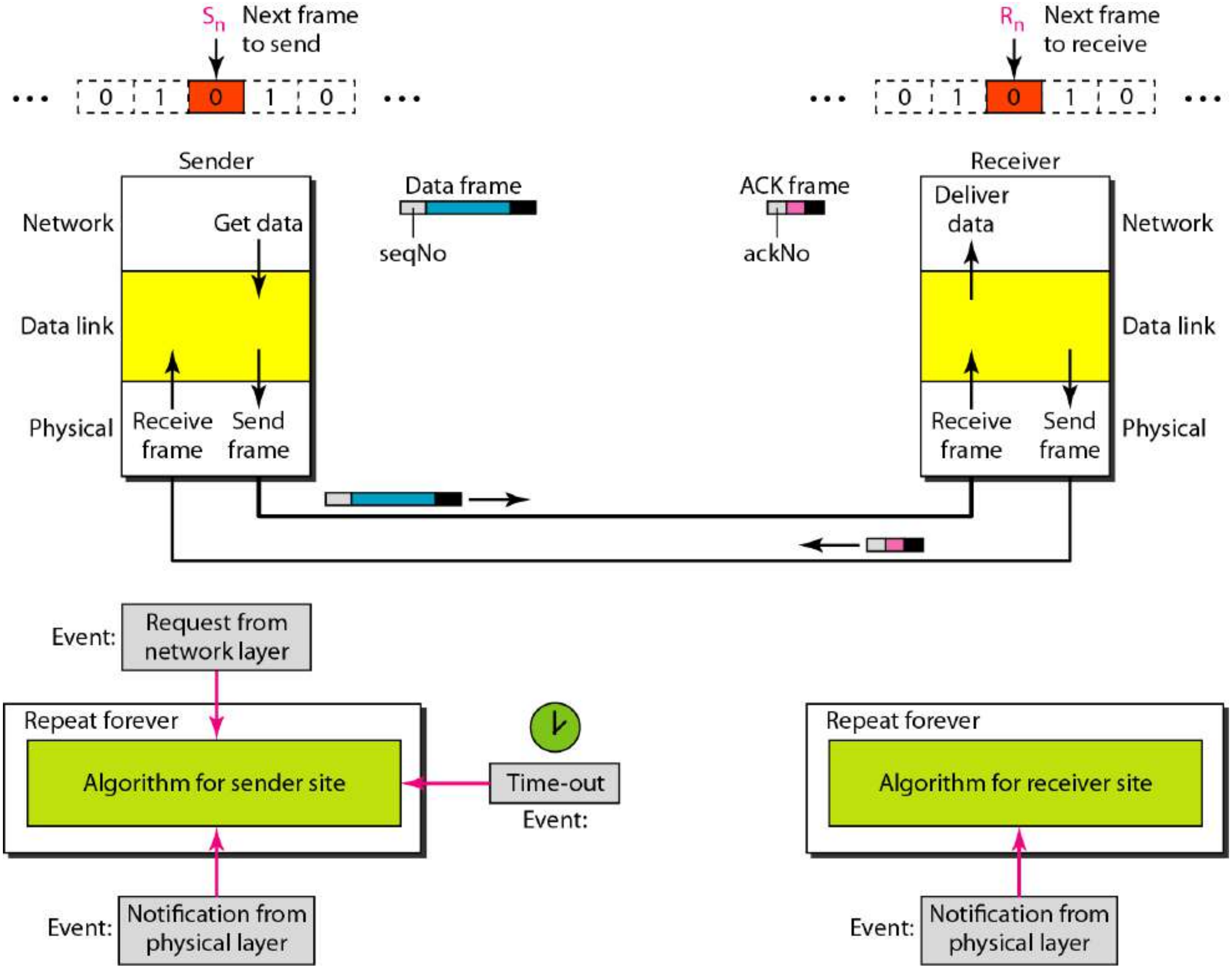
Noisy-Channel Protocols:

- Although the Stop-and-Wait Protocol gives us an idea of how to add flow control to its predecessor, noiseless channels are nonexistent. We discuss three protocols in this section that use error control.
  - ➢ Stop-and-Wait ARQ (Automatic Repeat Request)
  - ➢ Go-Back-N ARQ
  - ➢ Selective Repeat ARQ

Stop-and-Wait ARQ (Automatic Repeat Request):

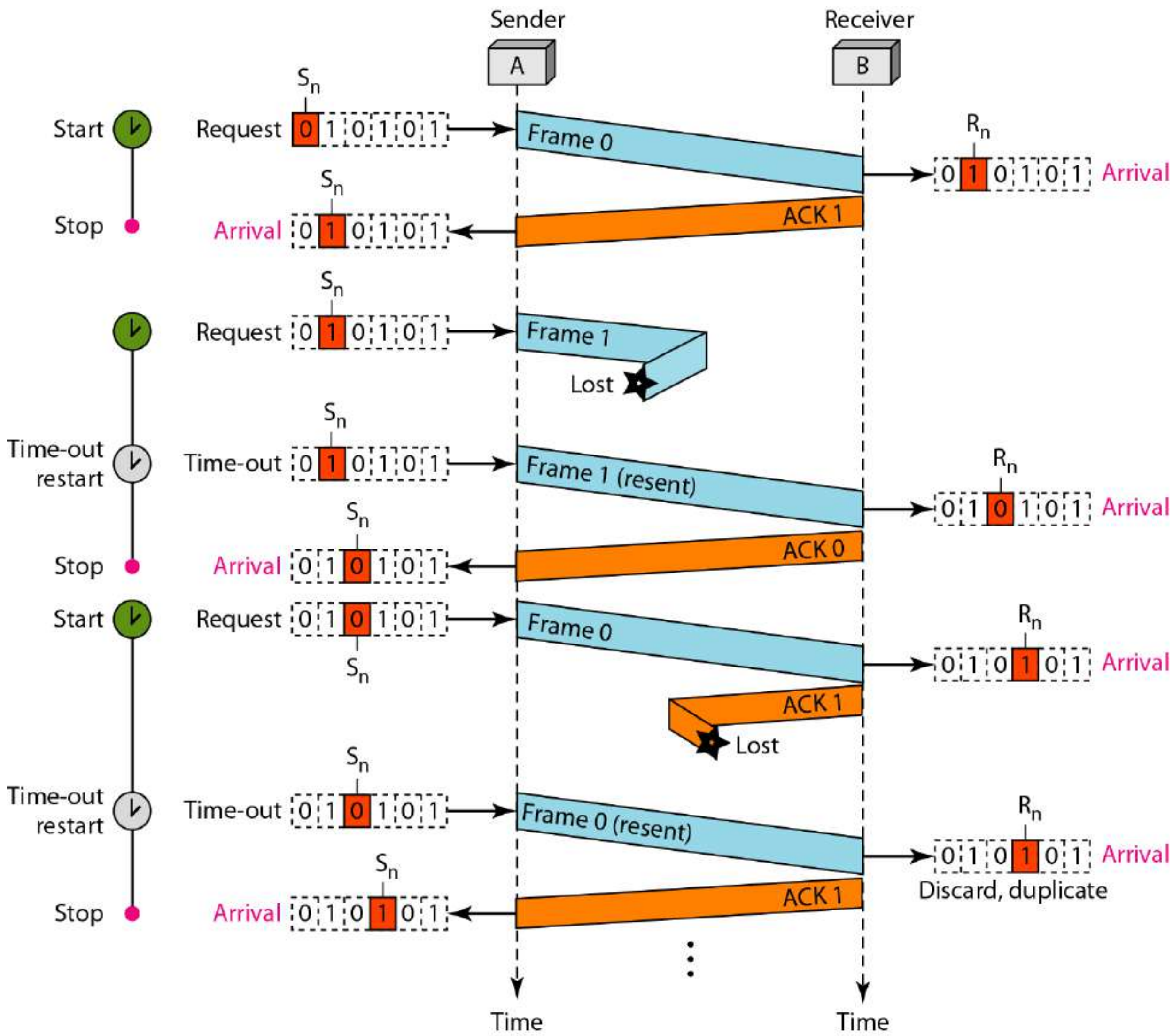
- Stop and wait ARQ adds a simple error control mechanism to stop and wait protocol.
- Error correction in Stop-and-Wait ARQ is done by keeping a copy of the sent frame and retransmitting of the frame when the timer expires.
- In Stop-and-Wait ARQ, we use **sequence numbers and acknowledge numbers** to number the frames**,** generally in stop and wait ARQ the sequence numbers are binary 0 or 1.
- **Design of stop and wait ARQ:**

the design of the Stop-and-Wait ARQ Protocol. The sending device keeps a copy of the last frame transmitted until it receives an acknowledgment for that frame. A data frames uses a seqNo (sequence number); an ACK frame uses an ackNo (acknowledgment number). The sender has a control variable, which we call S_n (sender, next frame to send), that holds the sequence number for the next frame to be sent (0 or 1).



Flow diagram of Stop and Wait ARQ:

Bellow figure shows an example of Stop-and-Wait ARQ. Frame 0 is sent and acknowledged. Frame 1 is lost and resent after the time-out. The resent frame 1 is acknowledged and the timer stops. Frame 0 is sent and acknowledged, but the acknowledgment is lost. The sender has no idea if the frame or the acknowledgment is lost, so after the time-out, it resends frame 0, which is acknowledged.



Go-Back-N Automatic Repeat Request (ARQ):

- To improve the efficiency of transmission (filling the pipe), multiple frames must be in transition while waiting for acknowledgment.
- In Go-Back-N Automatic Repeat Request, we can send several frames before receiving acknowledgments; we keep a copy of these frames until the acknowledgments arrive.
- ***Sequence Numbers:***
  Frames from a sending station are numbered sequentially. If the header of the frame allows m bits for the sequence number, the sequence numbers range from 0 to 2^m- 1. For example, if m is 4, the only sequence numbers are 0 through 15 inclusive. However, we can repeat the sequence. So the sequence numbers are

  0, 1,2,3,4,5,6, 7,8,9, 10, 11, 12, 13, 14, 15,0, 1,2,3,4,5,6,7,8,9,10, 11, ...

  In other words, the sequence numbers are modulo-2 m.

- ***Sliding Window:***
  In this protocol (and the next Selective Repeat ARQ), the sliding window is an abstract concept that defines the range of sequence numbers that is the concern of the sender and receiver.
  In other words, the sender and receiver need to deal with only part of the possible sequence numbers. The range which is the concern of the sender is called the send sliding window; the range that is the concern of the receiver is called the receiver sliding window.
  The send window is an imaginary box covering the sequence numbers of the data frames which can be in transit.
  In each window position, some of these sequence numbers define the frames that have been sent; others define those that can be sent. The maximum size of the window is $2^m – 1$. The size can be fixed and set to the maximum value. The following figure shows a sliding window of size 15 (m =4).
- **Flow diagrams of Go Back N ARQ:**

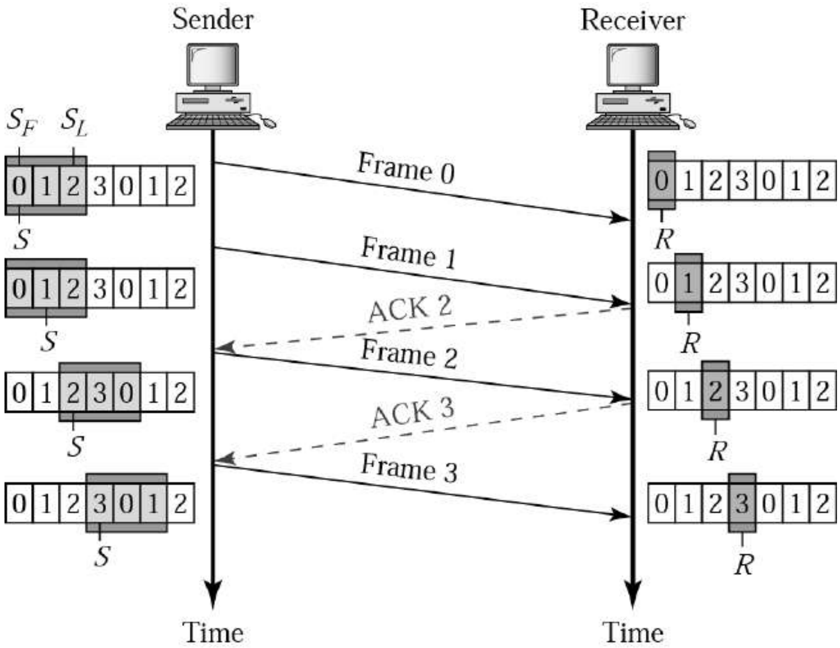


Figure: Go-Back-N ARQ Protocol Working flow (No loss of frames)

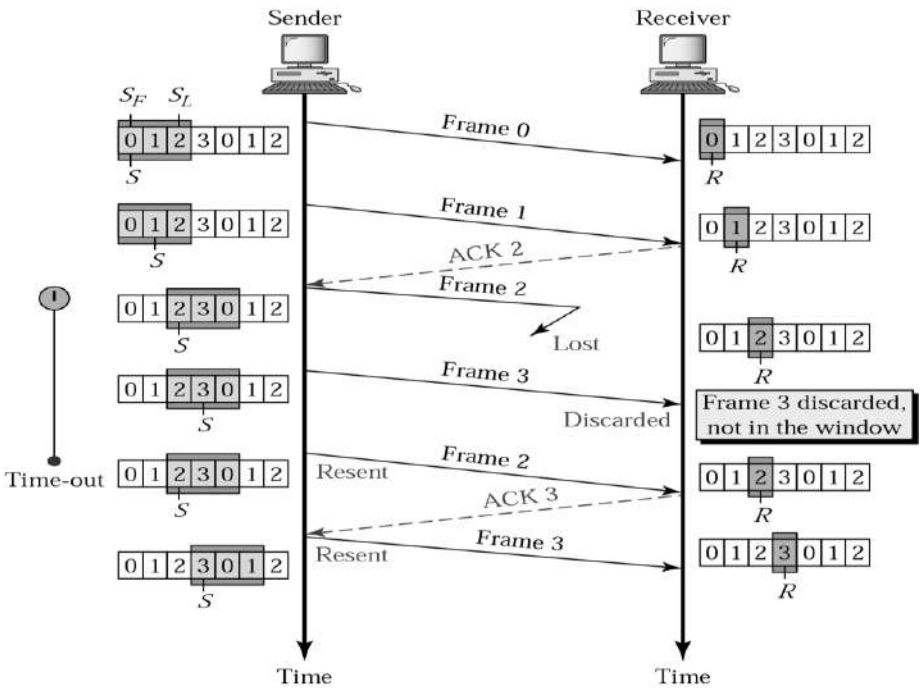


Figure: Go-back-N with lost frame

Selective Repeat ARQ Protocol:

- In this protocol rather than discard all the subsequent frames following a damaged or lost frame, the receiver's data link layer simply stores them in buffers.
- When the sender does not receive an acknowledgement for the first frame it's timer goes off after a certain time interval and it retransmits only the lost frame.
- Assuming error - free transmission this time, the sender's data link layer will have a sequence of a many correct frames which it can hand over to the network layer. Thus there is less overhead in retransmission than in the case of Go Back n protocol.
- In case of selective repeat protocol the window size may be calculated as follows. Assume that the size of both the sender's and the receiver's window is w. So initially both of them contain the values 0 to (w-1).
- Consider that sender's data link layer transmits all the w frames, the receiver's data link layer receives them correctly and sends acknowledgements for each of them. However, all the acknowledgments are lost and the sender does not advance it's window.
- The receiver window at this point contains the values w to (2w-1). To avoid overlap when the sender's data link layer retransmits, we must have the sum of these two windows less than sequence number space. Hence, we get the condition.

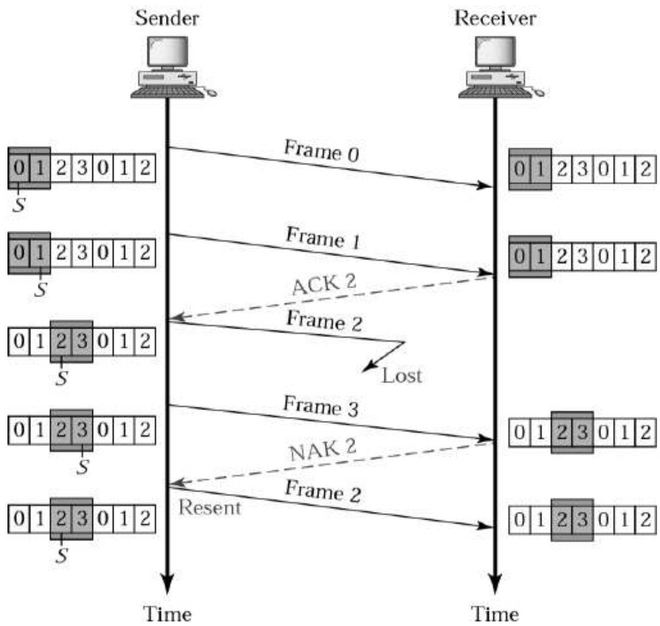


Figure: Selective-Repeat ARQ

Note: Both Go-Back-N and Selective Repeat ARQ protocols are called as Sliding window protocols

Piggybacking:

- A method to combine a data frame with ACK. Station A and B both have data to send.
- Instead of sending separately, station A sends a data frame that includes an ACK. Station B does the same thing.
- Piggybacking saves bandwidth.

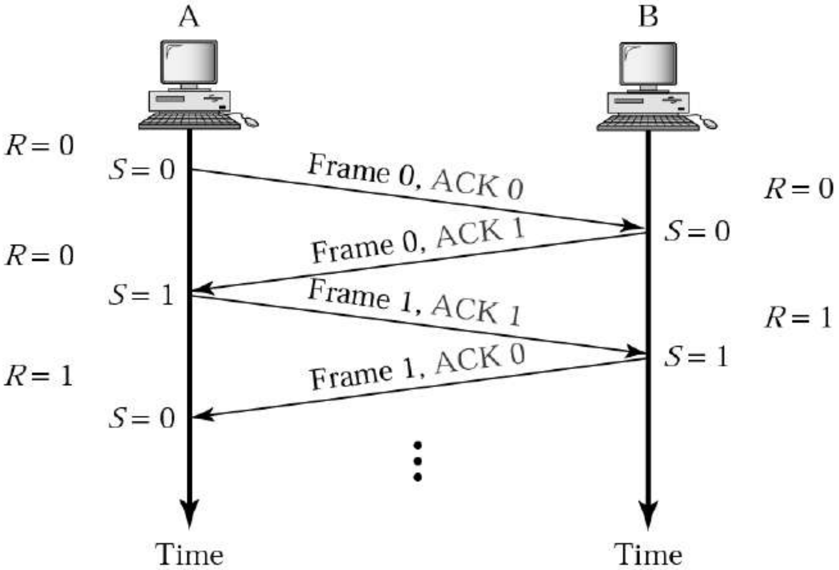
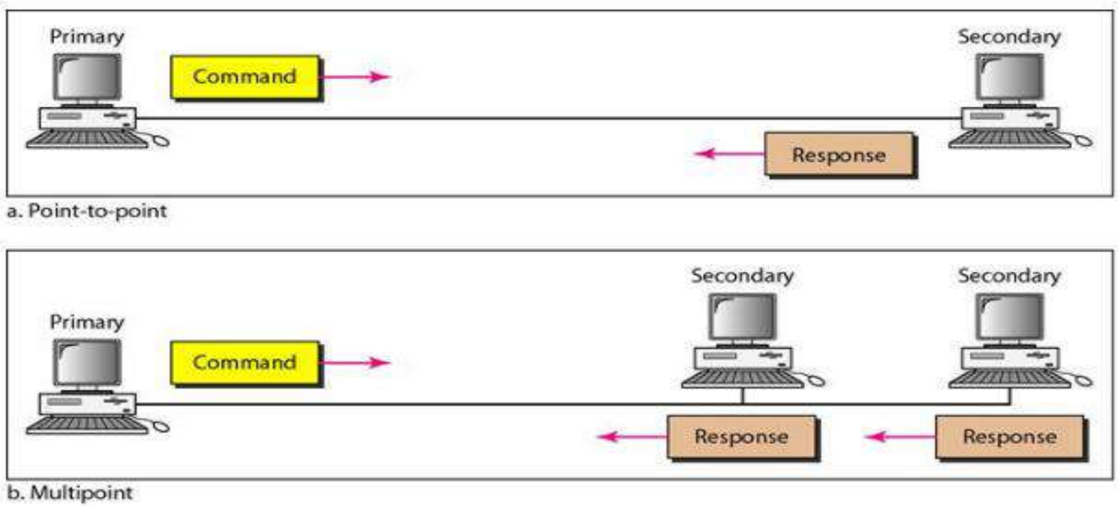


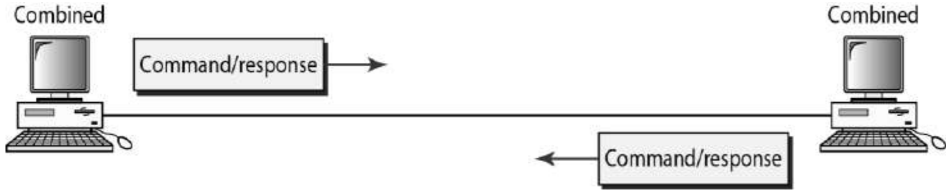
Figure: Piggybacking

High-level Data Link Control(HDLC):

- High-level Data Link Control (HDLC) is a bit-oriented protocol for communication over point-to-point and multipoint links. It implements the ARQ mechanisms.
- ***Configurations and Transfer Modes:***
  - HDLC provides two common transfer modes that can be used in different configurations:
    - Normal response mode (NRM) and
    - Asynchronous balanced mode (ABM).
  - **Normal Response Mode:**
    In normal response mode (NRM), the station configuration is unbalanced. 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, as shown in the following figure.

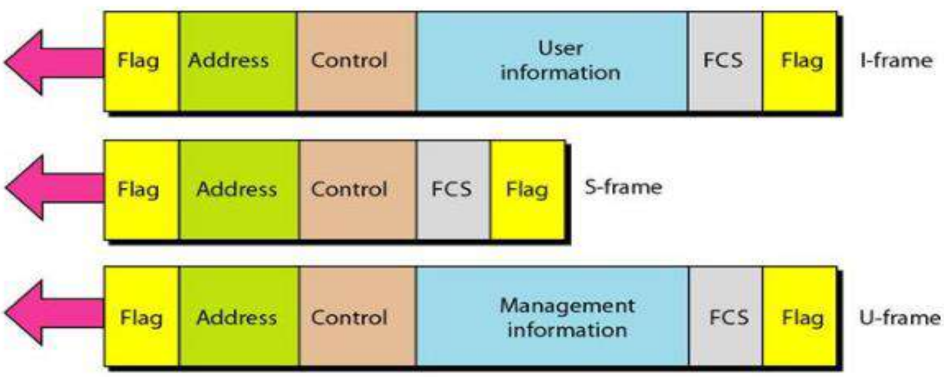


- **Asynchronous Balanced Mode:**
  In asynchronous balanced mode (ABM), the configuration is balanced. The link is point-to-point, and each station can function as a primary and a secondary (acting as peers). This is the common mode today.



Frames in HDLC:

- To provide the flexibility necessary to support all the options possible in the modes and configurations.
- HDLC defines three types of frames: Information frames (I-frames), Supervisory frames (S-frames), Unnumbered frames (U-frames). Each type of frame serves as an envelope for the transmission of a different type of message.
- **I-frames** are used to transport user data and control information relating to user data (piggybacking).
- **S-frames** are used only to transport control information.
- **U-frames** are reserved for system management. Information carried by U-frames is intended for managing the link itself.
- **Frame Format:**
  Each frame in HDLC may contain up to six fields, as shown in the following figure, a beginning flag field, an address field, a control field, an information field, a frame check sequence (FCS) field, and an ending flag field. In multiple-frame transmissions, the ending flag of one frame can serve as the beginning flag of the next frame.



- **Fields**
  1. **Flag field:** The flag field of an HDLC frame is an 8-bit sequence with the bit pattern 01111110 that identifies both the beginning and the end of a frame and serves as a synchronization pattern for the receiver.
  2. **Address field:** The second field of an HDLC frame 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. An address field can be 1 byte or several bytes long, depending on the needs of the network. One byte can identify up to 128 stations (l bit is used for another purpose). Larger networks require multiple-byte address fields. If the address field is only 1 byte, the last bit is always a 1. If the address is more than 1 byte, all bytes but the last one will end with 0; only the last will end with 1. Ending each intermediate byte with 0 indicates to the receiver that there are more address bytes to come.

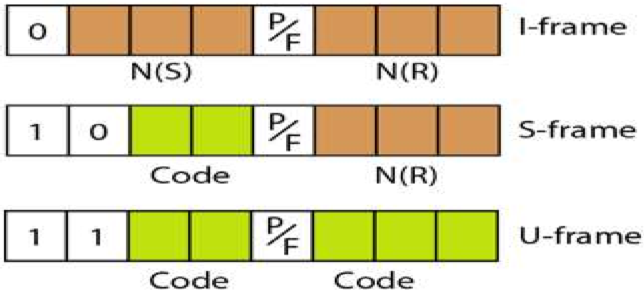
3. **Control field:** The control field is a 1- or 2-byte segment of the frame used for flow and error control. The interpretation of bits in this field depends on the frame type. We discuss this field later and describe its format for each frame type.

4. **Information field:** The information 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:** The frame check sequence (FCS) is the HDLC error detection field. It can contain either a 2- or 4-byte ITU-T CRC.

Control field:

- Control field in HDLC determines the type of frame and its functionality. The following diagram shows control fields for 3 different types of frames.



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 used to define these functions. The first bit defines the type. If the first bit of the control field is 0, this means the frame is an I-frame. The next 3 bits, called *N(S)*, define the sequence number of the frame. Note that with 3 bits, we can define a sequence number between 0 and 7; but in the extension format, in which the control field is 2 bytes, this field is larger. The last 3 bits, called *N(R)*, correspond to the acknowledgment number when piggybacking is used. 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 has meaning only when it is set (bit = 1) and can mean poll or final. It means *poll* when the frame is sent by a primary station to a secondary (when the address field contains the address of the receiver). It means *final* when the frame is sent by a secondary to a primary (when the address field contains the address of the sender).

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. If the first 2 bits of the control field is 10, this means the frame is an S-frame. The last 3 bits, called *N(R)*, corresponds to the acknowledgment number (ACK) or negative acknowledgment number (NAK) depending on the type of S-frame. 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, as listed below

- RR: Receive Ready
- RNR: Receive Not Ready
- REJ: Reject
- SREJ: Selective Reject

Control Field for U-Frames

Unnumbered frames are used to exchange session management and control information between connected devices. Unlike S-frames, U-frames contain an information field, but one used for system management information, not user data. As with S-frames, however, much of the information carried by U-frames is contained in codes included in the control field. U-frame codes are divided into two sections: a 2-bit prefix before the P/F bit and 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.

Point-to-Point protocol:

- Although HDLC is a general protocol that can be used for both point-to-point and multipoint configurations, one of the most common protocols for point-to-point access is the Point-to-Point Protocol (PPP). PPP is a byte-oriented protocol.

Services of PPP:

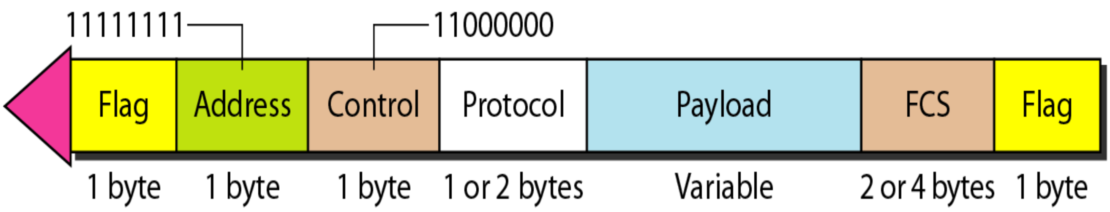
1. PPP defines the format of the frame to be exchanged between devices.
2. PPP defines how two devices can negotiate the establishment of the link and the exchange of data.
3. PPP defines how network layer data are encapsulated in the data link frame.
4. PPP defines how two devices can authenticate each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

Missing Services of PPP:

1. PPP does not provide flow control. A sender can send several frames one after another with no concern about overwhelming the receiver.
2. PPP has a very simple mechanism for error control. A CRC field is used to detect errors. If the frame is corrupted, it is silently discarded; the 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. PPP does not provide a sophisticated addressing mechanism to handle frames in a multipoint configuration.

Framing in PPP:

- PPP is a byte-oriented protocol, the frame format of PPP is as follows



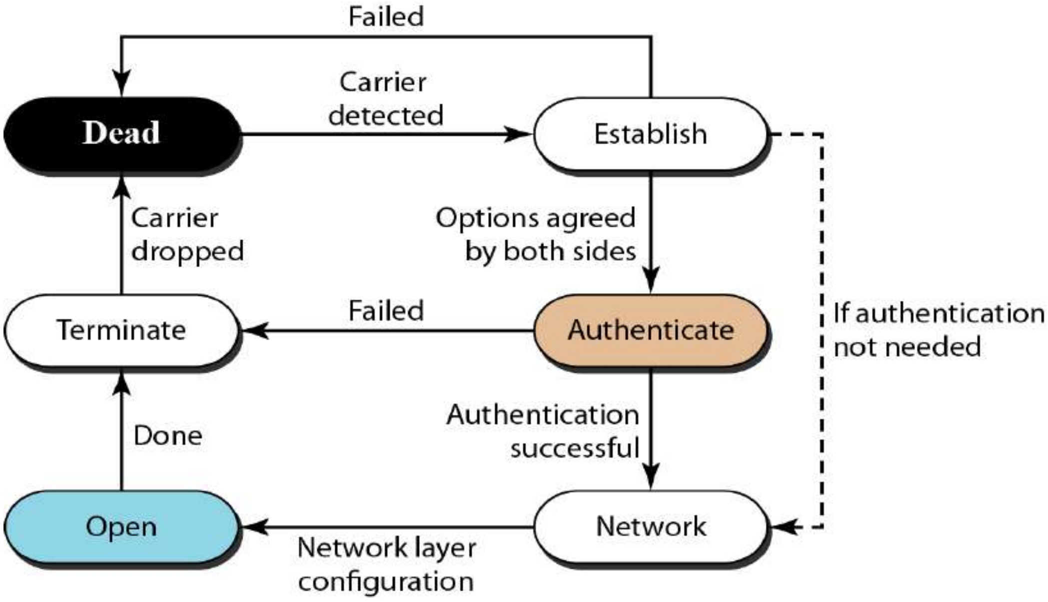
- **Flag.** A PPP frame starts and ends with a 1-byte flag with the bit pattern 01111110. Although this pattern is the same as that used in HDLC, there is a big difference. PPP is a byte-oriented protocol; HDLC is a bit-oriented protocol. The flag is treated as a byte, as we will explain later.
- **Address.** The address field in this protocol is a constant value and set to 11111111 (broadcast address). During negotiation (discussed later), the two parties may agree to omit this byte.
- **Control.** This field is set to the constant value 11000000 (imitating unnumbered frames in HDLC). As we will discuss later, PPP does not provide any flow control. Error control is also limited to error detection. This means that this field is not needed at all, and again, the two parties can agree, during negotiation, to omit this byte.
- **Protocol.** The protocol field defines what is being carried in the data field: either user data or other information. We discuss this field in detail shortly. This field is by default 2 bytes long, but the two parties can agree to use only 1 byte.
- **Payload field.** This field carries either the user data or other information that we will discuss shortly. The data field is a sequence of bytes with the default of a maximum of 1500 bytes; but this can be changed during negotiation. The data field is byte-stuffed if the flag byte pattern appears in this field. Because there is no field defining the size of the data field, padding is needed if the size is less than the maximum default value or the maximum negotiated value.
- **FCS.** The frame check sequence (FCS) is simply a 2-byte or 4-byte standard CRC.

Byte Stuffing

The similarity between PPP and HDLC ends at the frame format. PPP, as we discussed before, is a byte-oriented protocol totally different from HDLC. As a byte-oriented protocol, the flag in PPP is a byte and 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 flaglike pattern appears in the data, this extra byte is stuffed to tell the receiver that the next byte is not a flag.

PPP is a byte-oriented protocol using byte stuffing with the escape byte 01111101.

Transition phases in PPP:



- **Dead.** In the dead phase the link is not being used. There is no active carrier (at the physical layer) and the line is quiet.
- **Establish.** When one of the nodes starts the communication, the connection goes into this phase. In this phase, options are negotiated between the two parties. If the negotiation is successful, the system goes to the authentication phase (if authentication is required) or directly to the networking phase. The link control protocol packets, discussed shortly, are used for this purpose. Several packets may be exchanged here.
- **Authenticate.** The authentication phase is optional; the two nodes may decide, during the establishment phase, not to skip this phase. However, if they decide to proceed with authentication, they send several authentication packets, discussed later. If the result is successful, the connection goes to the networking phase; otherwise, it goes to the termination phase.
- **Network.** In the network phase, negotiation for the network layer protocols takes place. PPP specifies that two nodes establish a network layer agreement before data at the network layer can be exchanged. The reason is that PPP supports multiple protocols at the network layer. If a node is running multiple protocols simultaneously at the network layer, the receiving node needs to know which protocol will receive the data.
- **Open.** In the open phase, data transfer takes place. When a connection reaches this phase, the exchange of data packets can be started. The connection remains in this phase until one of the endpoints wants to terminate the connection.
- **Terminate.** In the termination phase the connection is terminated. Several packets are exchanged between the two ends for house cleaning and closing the link.

Multiplexing in PPP:

Although PPP is a data link layer protocol, PPP uses another set of other 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). At any moment, a PPP packet can carry data from one of these protocols in its data field,

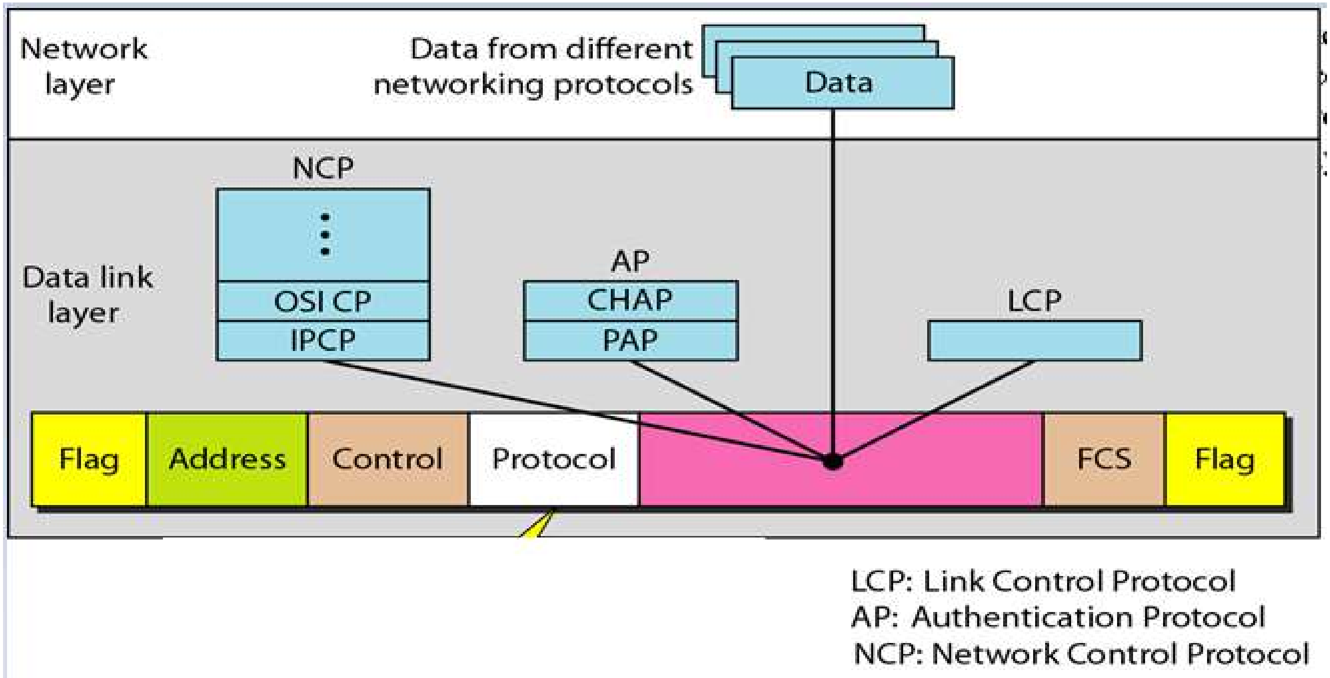


Figure: Multiplexing in PPP

Link Control Protocol

The **Link Control Protocol (LCP)** is responsible for establishing, maintaining, configuring, and terminating links. It also provides negotiation mechanisms to set options between the two endpoints. Both endpoints of the link must reach an agreement about the options before the link can be established.

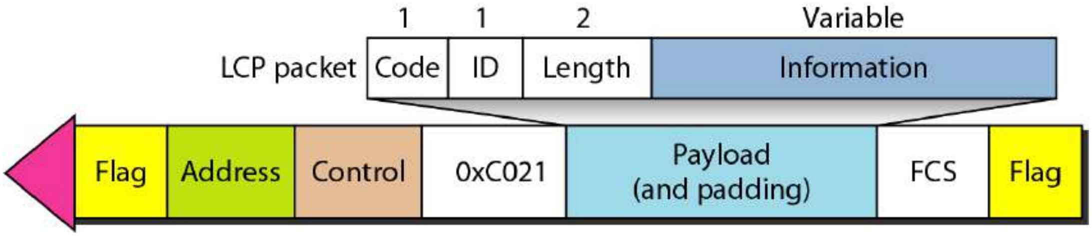


Figure: LCP Packet format

Authentication Protocols

Authentication plays a very important role in PPP because PPP is designed for use over dial-up links where verification of user identity is necessary. **Authentication** means validating the identity of a user who needs to access a set of resources. PPP has created two protocols for authentication: Password Authentication Protocol and Challenge Handshake Authentication Protocol. Note that these protocols are used during the authentication phase.

PAP The **Password Authentication Protocol (PAP)** is a simple authentication procedure with a two-step process:

1. The user who wants to access a system sends an authentication identification (usually the user name) and a password.
2. The system checks the validity of the identification and password and either accepts or denies connection.

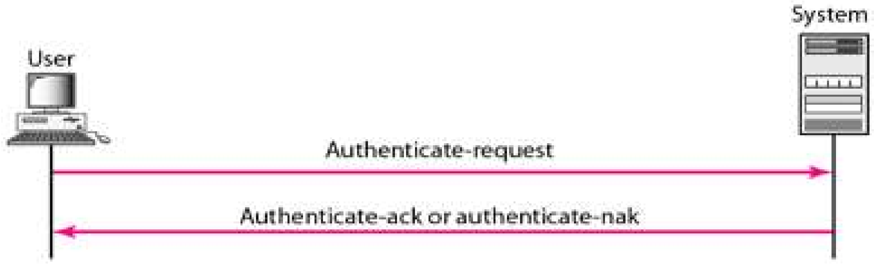


Figure: PAP

CHAP The **Challenge Handshake Authentication Protocol (CHAP)** is a three-way hand-shaking authentication protocol that provides greater security than PAP. In this method, the password is kept secret; it is never sent online.

1. The system sends the user a challenge packet containing a challenge value, usually a few bytes.
2. The user applies a predefined function that takes the challenge value and the user's own password and creates a result. The user sends the result in the response packet to the system.
3. The system does the same. It applies the same function to the password of the user (known to the system) and the challenge value to create a result. If the result created is the same as the result sent in the response packet, access is granted; otherwise, it is denied. CHAP is more secure than PAP, especially if the system continuously changes the challenge value.

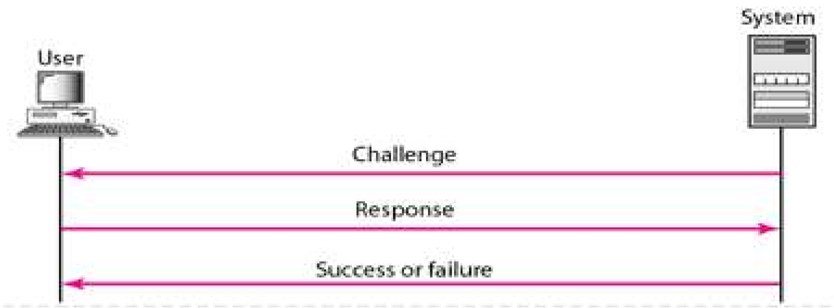


Figure: CHAP

Network Control Protocols

PPP is a multiple-network layer protocol. It can carry a network layer data packet from protocols defined by the Internet, OSI, Xerox, DECnet, AppleTalk, Novel, and so on.

To do this, PPP has defined a specific Network Control Protocol for each network protocol. For example, IPCP (Internet Protocol Control Protocol) configures the link for carrying IP data packets. Xerox CP does the same for the Xerox protocol data packets, and so on. Note that none of the NCP packets carry network layer data; they just configure the link at the network layer for the incoming data.

IPCP One NCP protocol is the **Internet Protocol Control Protocol (IPCP).** This protocol configures the link used to carry IP packets in the Internet. IPCP is especially of interest to us. The format of an IPCP packet is shown in Figure

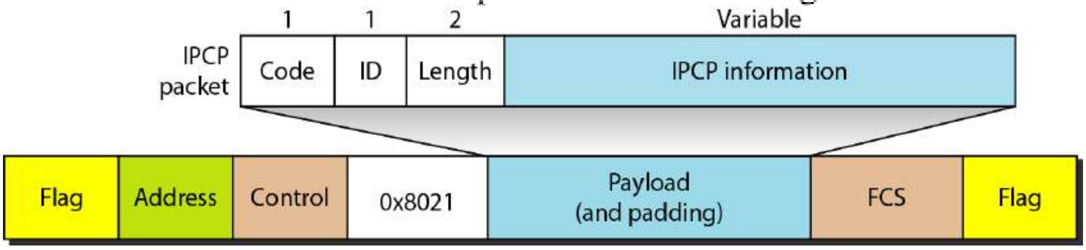


Figure: IPCP format

Data from the Network Layer

After the network layer configuration is completed by one of the NCP protocols, the users can exchange data packets from the network layer. Here again, there are different protocol fields for different network layers. For example, if PPP is carrying data from the IP network layer, the field value is 0021 (note that the three rightmost digits are the same as for IPCP). If PPP is carrying data from the OSI network layer, the value of the protocol field is 0023, and so on.

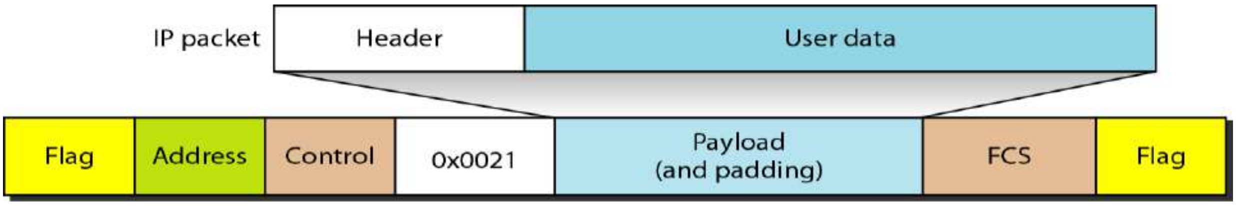


Figure: ***IP datagram encapsulated in a PPP frame***

Multilink in PPP:

PPP was originally designed for a single-channel point-to-point physical link. The availability of multiple channels in a single point-to-point link motivated the development of Multilink PPP. In this case, a logical PPP frame is divided into several actual PPP frames. A segment of the logical frame is carried in the payload of an actual PPP frame,

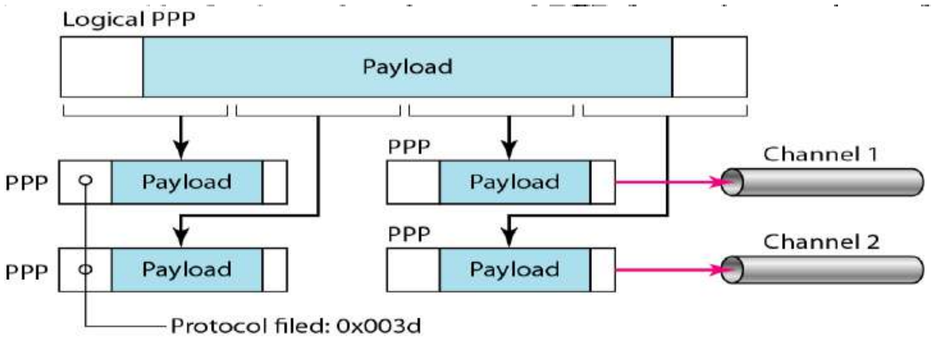


Figure: Multilink PPP

****