UNIT – 2

DATA LINK LAYER DESIGN ISSUE

Unit-02/Lecture-01

FRAMING[RGPV/Dec 2008,Dec 2011,Dec 2013, Jun 2014]

Data transmission in the physical layer means moving bits in the form of a signal from the source to the destination. The physical layer provides bit synchronization to ensure that the sender and receiver use the same bit durations and timing.

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. In addition, each envelope defines the sender and receiver addresses since the postal system is a many-to-many carrier facility.

Framing in the data link layer separates a message from one source to a destination, or from other messages to other destinations, by adding a sender address and a destination address. The destination address defines where the packet is to go; the sender address helps the recipient acknowledge the receipt.

Although the whole message could be packed in one frame that is not normally done. One reason is that a frame can be very large, making flow and error control very inefficient. When a message is carried in one very large frame, even a single-bit error would require the retransmission of the whole message. When a message is divided into smaller frames, a single-bit error affects only that small frame.

Fixed-Size Framing

Frames can be of fixed or variable size. In fixed-size framing, there is no need for defining the boundaries of the frames; the size itself can be used as a delimiter. An example of this type of framing is the ATM wide-area network, which uses frames of fixed size called cells.

Variable-Size Framing

In variable-size framing, we need a way to define the end of the frame and the beginning of the next. Historically, two approaches were used for this purpose:

A character-oriented approach and a bit-oriented approach.

Character-Oriented Protocols

In a character-oriented protocol, data to be carried are 8-bit characters from a coding system. The header, which normally carries the source and destination addresses and other control information, and the trailer, which carries error detection or error correction redundant bits, are also multiples of 8 bits. To separate one frame from the next, an 8-bit (I-byte) flag is added at the beginning and the end of a frame. The flag, composed of protocol-dependent special characters, signals the start or end of a frame.

The flag could be selected to be any character not used for text communication. Now, however, we send other types of information such as graphs, audio, and video. Any pattern used for the flag could also be part of the information. If this happens, the receiver, when it encounters this pattern in the middle of the data, thinks it has reached the end of the frame. To fix this problem, a byte-stuffing strategy was added to character-oriented framing. In byte stuffing (or character 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 usually called the escape character (ESC), which has a predefined bit pattern. Whenever the receiver encounters the ESC character, it removes it from the data section and treats the next character as

data, not a delimiting flag.

Byte stuffing by the escape character allows the presence of the flag in the data section of the frame, but it creates another 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. To solve this problem, the escape characters that are part of the text must also be marked by another escape character. In other words, if the escape character is part of the text, an extra one is added to show that the second one is part of the text.

Byte stuffing is the process of adding 1 extra byte whenever there is a flag or escape character in the text.

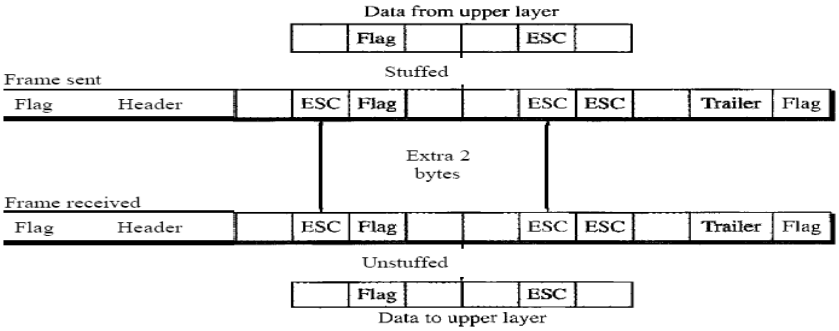

FIGURE 2.1: BYTE STUFFING

Bit-Oriented Protocols[RGPV/Dec 2007]

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

This flag can create the same type of problem we saw in the byte-oriented protocols. That is, if the flag pattern appears in the data, we need to somehow inform the receiver that this is not the end of the frame. We do this by stuffing 1 single bit (instead of I byte) to prevent the pattern from looking like a flag. The strategy is called bit stuffing. In bit stuffing, if a 0 and five consecutive I bits are encountered, an extra 0 is added. This extra stuffed bit is eventually removed from the data by the receiver. Note that the extra bit is added after one 0 followed by five 1s regardless of the value of the next bit. This guarantees that the flag field sequence does not inadvertently appear in the frame.

Bit stuffing is the process of adding one extra 0 whenever five consecutive 18 follow a 0 in the data, so that the receiver does not mistake the pattern 0111110 for a flag.

Bit stuffing at the sender and bit removal at the receiver. Note that even if we have a 0 after five 1s, we still stuff a 0. The 0 will be removed by the receiver.

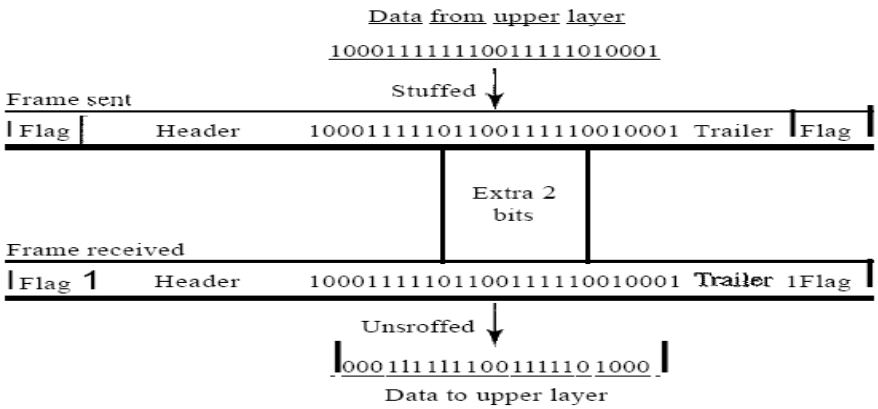

FIGURE 2.2: BIT STUFFING

FLOW AND ERROR CONTROL [RGPV/Jun 2014]

Data communication requires at least two devices working together, one to send and the other to receive. Even such a basic arrangement requires a great deal of coordination for an intelligible exchange to occur. The most important responsibilities of the data link layer are flow control and error control. Collectively, these functions are known as data link control.

Flow Control

Flow control coordinates the amount of data that can be sent before receiving an acknowledgment and is one of the most important duties of the data link layer. In most protocols, flow control is a set of procedures that tells the sender how much data it can transmit before it must wait for an acknowledgment from the receiver. The flow of data must not be allowed to overwhelm the receiver. Any receiving device has a limited speed at which it can process incoming data and a limited amount of memory in which to store incoming data. The receiving device must be able to inform the sending device before those limits are reached and to request that the transmitting device send fewer frames or stop temporarily. Incoming data must be checked and processed before they can be used. The rate of such processing is often slower than the rate of transmission. For this reason, each receiving device has a block of memory, called a buffer, reserved for storing incoming data until they are processed. If the buffer begins to fill up, the receiver must be able to tell the sender to halt transmission until it is once again able to receive.

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

Error Control

Error control is both error detection and error correction. It allows the receiver to inform the sender of any frames lost or damaged in transmission and coordinates the retransmission of those frames by the sender. In the data link layer, the term error control refers primarily to methods of error detection and retransmission. Error control in the data link layer is often implemented simply: Any time an error is detected in an exchange, specified frames are retransmitted. This process is called automatic repeat request (ARQ).

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

S.NO	RGPV QUESTIONS	Year	Marks
Q.1	Discuss the data link layer design issue.	Dec 2010 Dec 2013	7
Q.2	Name important functions of data link layer.	Dec 2006 Jun 2010	7
Q.3	Write a detail note on framing.	Dec 2011	7
Q.4	If start and end header is 100001 and the following data stream is to be bit	Dec 2007	7

we dont take any liability for the notes correctness. http://www.rgpvonline.com

	stuffed 1001100011000011100000011 What will be the frame after bit stuffing?		
Q.5	Data link layer in computer networks address framing, flow control and error control issues. Explain why these functions are important and how they are implemented.	Jun 2014	7

we dont take any liability for the notes correctness. http://www.rgpvonline.com

DATA LINK LAYER PROTOCOLS

Unit-02/Lecture-02

Protocols

- Simplest
- Stop-and-Wait
- Stop-and-Wait ARQ
- Go-Hack-N ARQ
- Selective Repeat ARQ

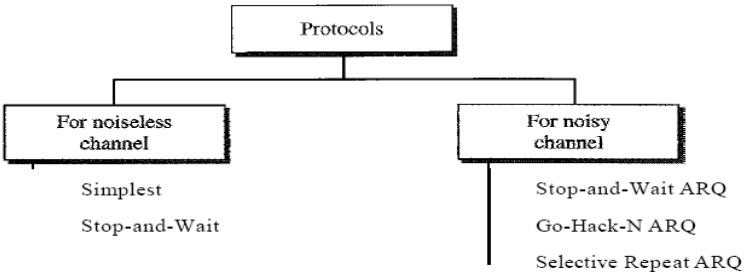

FIGURE 2.3: DATA LINK LAYER PROTOCOL

All the protocols unidirectional in the sense that the data frames travel from one node, called the sender, to another node, called the receiver. Although special frames, called acknowledgment (ACK) and negative acknowledgment (NAK) can flow in the opposite direction for flow and error control purposes, data flow in only one direction.

In a real-life network, the data link protocols are implemented as bidirectional; data flow in both directions. In these protocols the flow and error control information such as ACKs and NAKs is included in the data frames in a technique called piggybacking. Because bidirectional protocols are more complex than unidirectional ones.

NOISELESS CHANNELS[RGPV/ Jun 2014]

We have an ideal channel in which no frames are lost, duplicated, or corrupted. We introduce two protocols for this type of channel. The first is a protocol that does not use flow control; the second is the one that does. Neither has error control because we have assumed that the channel is a perfect noiseless channel.

- **Simplest Protocol** It has no flow or error control. It is a unidirectional protocol in which data frames are travelling in only one direction-from the sender to receiver. We assume that the receiver can immediately handle any frame it receives with a processing time that is small enough to be negligible. The data link layer of the receiver immediately removes the header from the frame and hands the data packet to its network layer, which can also accept the packet immediately. In other words, the receiver can never be overwhelmed with incoming frames.

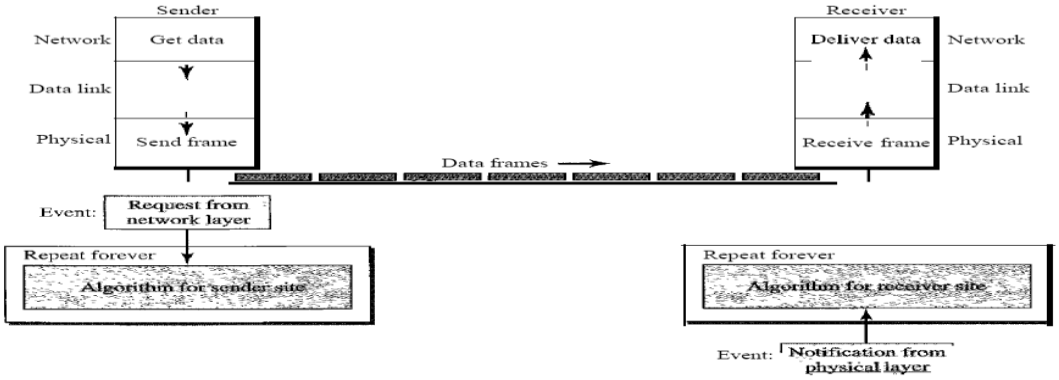

FIGURE 2.4: SIMPLEST PROTOCOL

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 signalling, multiplexing, and so on) for the physical transmission of bits.

The sender site cannot send a frame until its network layer has a data packet to send. The receiver site cannot deliver a data packet to its network layer until a frame arrives. If the protocol is implemented as a procedure, we need to introduce the idea of events in the protocol. The procedure at the sender site is constantly running; there is no action until there is a request from the network layer. The procedure at the receiver site is also constantly running, but there is no action until notification from the physical layer arrives. Both procedures are Constantly running because they do not know when the corresponding events will occur.

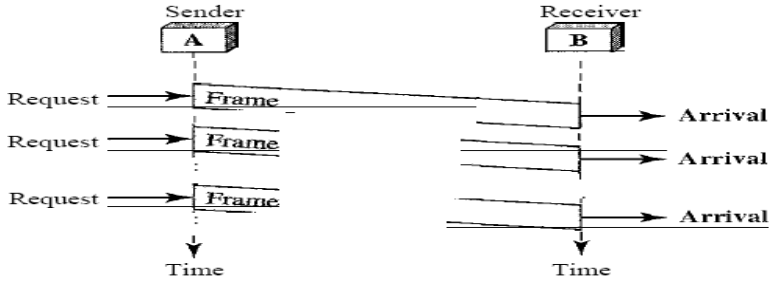

FIGURE 2.5: SIMPEST PROTOCOL

- **Stop-and-Wait Protocol[RGPV/Jun 2010, Jun 2011]**

  If data frames arrive at the receiver site faster than they can be processed, the frames must be stored until their use. Normally, the receiver does not have enough storage space, especially if it is receiving data from many sources. This may result in either the discarding of frames or denial of service. To prevent the receiver from becoming overwhelmed with frames, we somehow need to tell the sender to slow down. There must be feedback from the receiver to the sender.

  The protocol is called the Stop-and-Wait Protocol because the 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.

Design

We can see the traffic on the forward channel (from sender to receiver) and the reverse channel. At any time, there is either one data frame on the forward channel or one ACK frame on the reverse channel. We therefore need a half-duplex link.

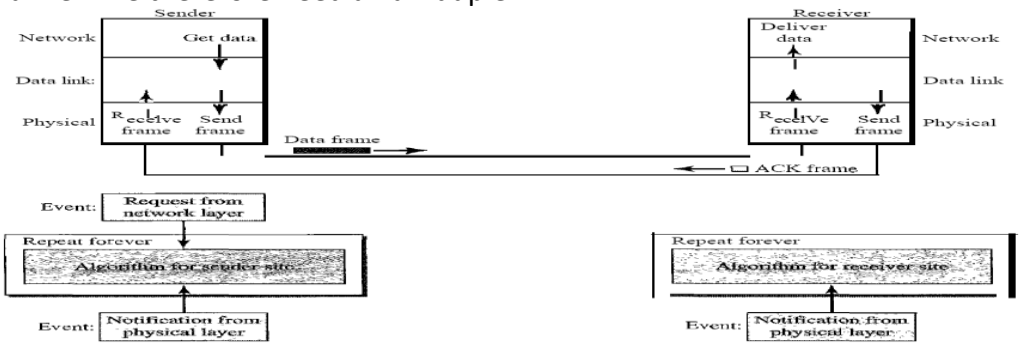

FIGURE 2.6: STOP & WAIT PROTOCOL

Analysis Here two events can occur: a request from the network layer or an arrival notification from the physical layer. The responses to these events must alternate. In other words, after a frame is sent, the algorithm must ignore another network layer request until that frame is

acknowledged. We know that two arrival events cannot happen one after another because the channel is error-free and does not duplicate the frames. The requests from the network layer, however, may happen one after another without an arrival event in between. We need somehow to prevent the immediate sending of the data frame. Although there are several methods, we have used a simple canSend variable that can either be true or false. When a frame is sent, the variable is set to false to indicate that a new network request cannot be sent until canSend is true. When an ACK is received, canSend is set to true to allow the sending of the next frame.

After the data frame arrives, the receiver sends an ACK frame (line 9) to acknowledge the receipt and allow the sender to send the next frame.

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.

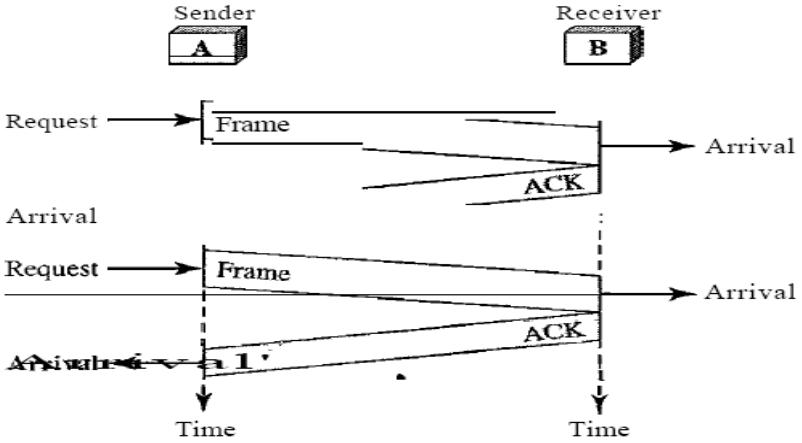

FIGURE 2.7: STOP & WAIT PROTOCOL FLOW DIAGRAM

S.NO	RGPV QUESTIONS	Year	Marks
Q.1	Explain stop & wait protocol.	June 2010 June 2011	7
Q.2	Write and explain window based flow control strategies.	Jun 2014	7

DATA LINK LAYER PROTOCOLS

Unit-02/Lecture-03

NOISY CHANNELS

Although the Stop-and-Wait Protocol gives us an idea of how to add flow control to its predecessor, noiseless channels are nonexistent. We can ignore the error (as we sometimes do), or we need to add error control to our protocols. We discuss three protocols in this section that use error control.

- **Stop-and-Wait Automatic Repeat Request[RGPV/Dec 2009, Jun 2013]**
  Our first protocol, called the Stop-and-Wait Automatic Repeat Request (Stop-and Wait ARQ), adds a simple error control mechanism to the Stop-and-Wait Protocol. Let us see how this protocol detects and corrects errors.

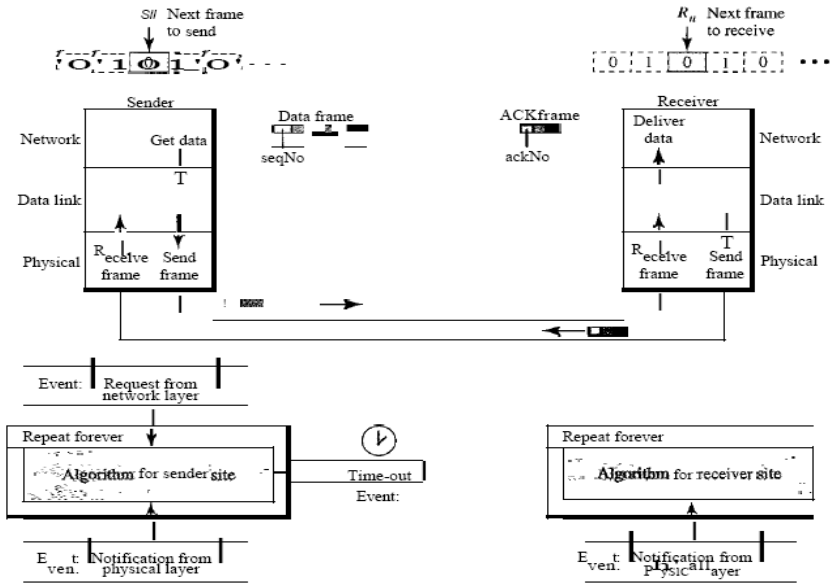

FIGURE 2.8: STOP & WAIT ARQ PROTOCOL

To detect and correct corrupted frames, we need to add redundancy bits to our data frame. When the frame arrives at the receiver site, it is checked and if it is corrupted, it is silently discarded. The detection of errors in this protocol is manifested by the silence of the receiver.

Lost frames are more difficult to handle than corrupted ones. In our previous protocols, there was no way to identify a frame. The received frame could be the correct one, or a duplicate, or a frame out of order. The solution is to number the frames. When the receiver receives a data frame that is out of order, this means that frames were either lost or duplicated.

The completed and lost frames need to be resent in this protocol. If the receiver does not respond when there is an error, how can the sender know which frame to resend? To remedy this problem, the sender keeps a copy of the sent frame. At the same time, it starts a timer. If the timer expires and there is no ACK for the sent frame, the frame is resent, the copy is held, and the timer is restarted. Since the protocol uses the stop-and-wait mechanism, there is only one specific frame that needs an ACK even though several copies of the same frame can be in the network.

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.

Since an ACK frame can also be corrupted and lost, it too needs redundancy bits and a sequence number. The ACK frame for this protocol has a sequence number field. In this protocol, the sender simply discards a corrupted ACK frame or ignores an out-of-order one.

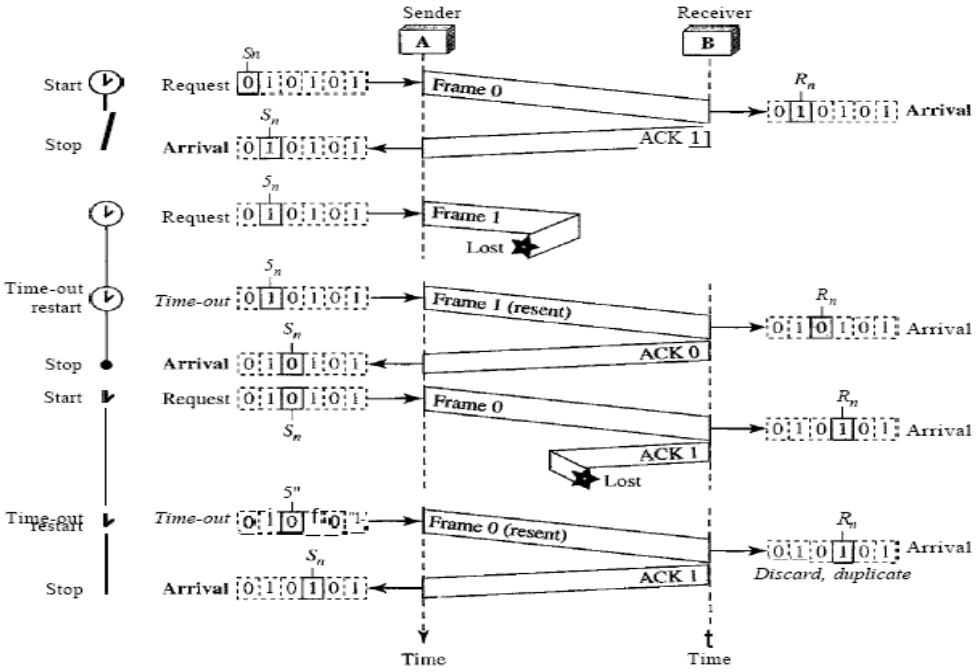

FIGURE 2.9: STOP & WAIT ARQ PROTOCOL FLOW DIAGRAM

Sequence Numbers

The protocol specifies that frames need to be numbered. This is done by using sequence numbers. A field is added to the data frame to hold the sequence number of that frame.

One important consideration is the range of the sequence numbers. Since we want to minimize the frame size, we look for the smallest range that provides unambiguous communication. The sequence numbers of course can wrap around. For example, if we decide that the field is m bits long, the sequence numbers start from 0, go to 2m - 1, and then are repeated.

Acknowledgment Numbers

Since the sequence numbers must be suitable for both data frames and ACK frames, we use this convention: The acknowledgment numbers always announce the sequence number of the next frame expected by the receiver. For example, if frame 0 has arrived safe and sound, the receiver sends an ACK frame with acknowledgment 1 (meaning frame 1 is expected next). If frame 1 has arrived safe and sound, the receiver sends an ACK frame with acknowledgment 0 (meaning frame 0 is expected).

Pipelining[RGPV/Jun 2010]

Improves the efficiency of the transmission if the number of bits in transition is large with respect to the bandwidth-delay product.

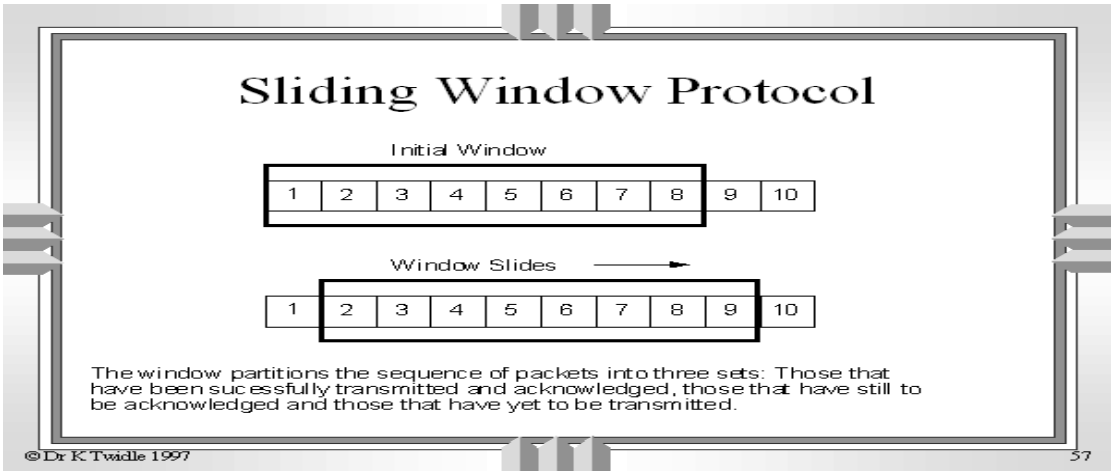

FIGURE 2.10: SLIDING WINDOW PROTOCOL

- **Go-Back-N Automatic Repeat Request[RGPV/Dec 2010, Jun 2013]**
  To improve the efficiency of transmission (filling the pipe), multiple frames must be in transition while waiting for acknowledgment. In other words, we need to let more than one frame be outstanding to keep the channel busy while the sender is waiting for acknowledgment.

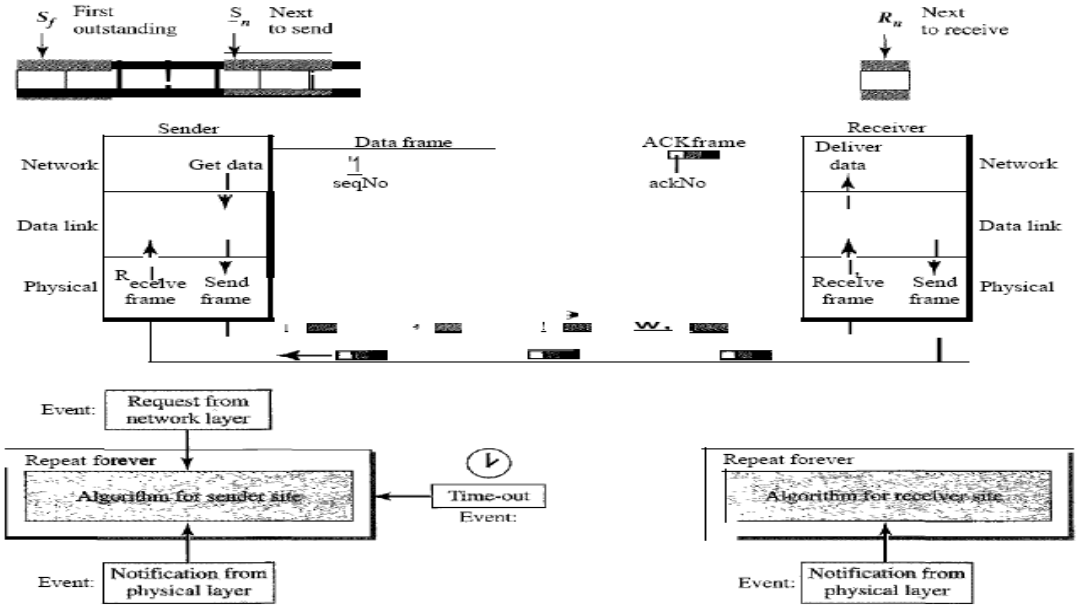

FIGURE 2.11: GO-BACK-N ARQPROTOCOL

The first is called Go-Back-N Automatic Repeat Request. In this protocol 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. However, because we need to include the sequence number of each frame in the header, we need to set a limit. If the header of the frame allows m bits for the sequence number, the sequence numbers range from 0 to 2m - 1. For example, if m is 4, the only sequence numbers are 0 through 15 inclusive.

The send window can slide one or more slots when a valid acknowledgment arrives.
The receive window makes sure that the correct data frames are received and that the correct acknowledgments are sent. The size of the receive window is always I. The receiver is always looking for the arrival of a specific frame. Any frame arriving out of order is discarded and needs to be resent.

The receive window also slides, but only one slot at a time. When a correct frame is received (and a frame is received only one at a time), the window slides.

Timers
Although there can be a timer for each frame that is sent, in our protocol we use only one. The reason is that the timer for the first outstanding frame always expires first; we send all outstanding frames when this timer expires.

Acknowledgment
The receiver sends a positive acknowledgment if a frame has arrived safe and sound and in order. If a frame is damaged or is received out of order, the receiver is silent and will discard all subsequent frames until it receives the one it is expecting. The silence of the receiver causes the timer of the unacknowledged frame at the sender site to expire.

This, in turn, causes the sender to go back and resend all frames, beginning with the one with the expired timer. The receiver does not have to acknowledge each frame received.
It can send one cumulative acknowledgment for several frames. Resending a

Frame

When the timer expires, the sender resends all outstanding frames. For example, suppose the sender has already sent frame 6, but the timer for frame 3 expires. This means that frame 3 has not been acknowledged; the sender goes back and sends frames 3, 4, 5, and 6 again. That is why the protocol is called Go-Back-N ARQ.

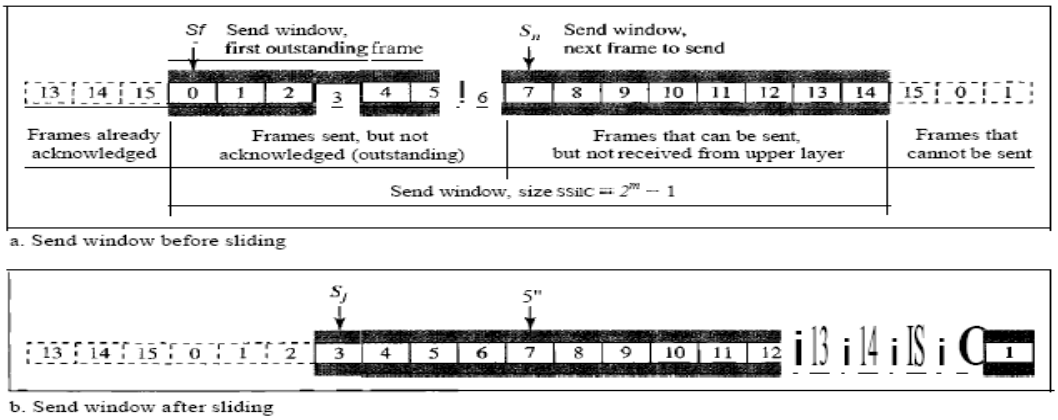

FIGURE 2.12: SENT WINDOW

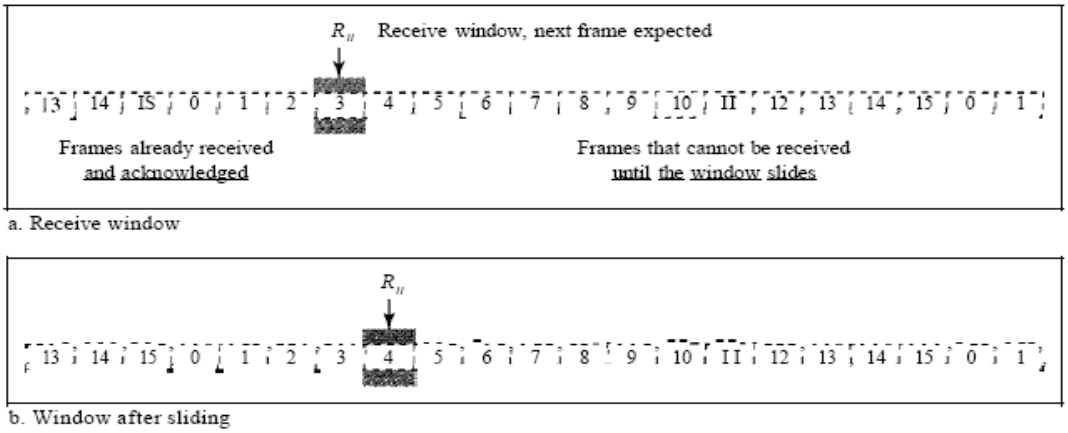

FIGURE 2.13: RECIEVER WINDOW

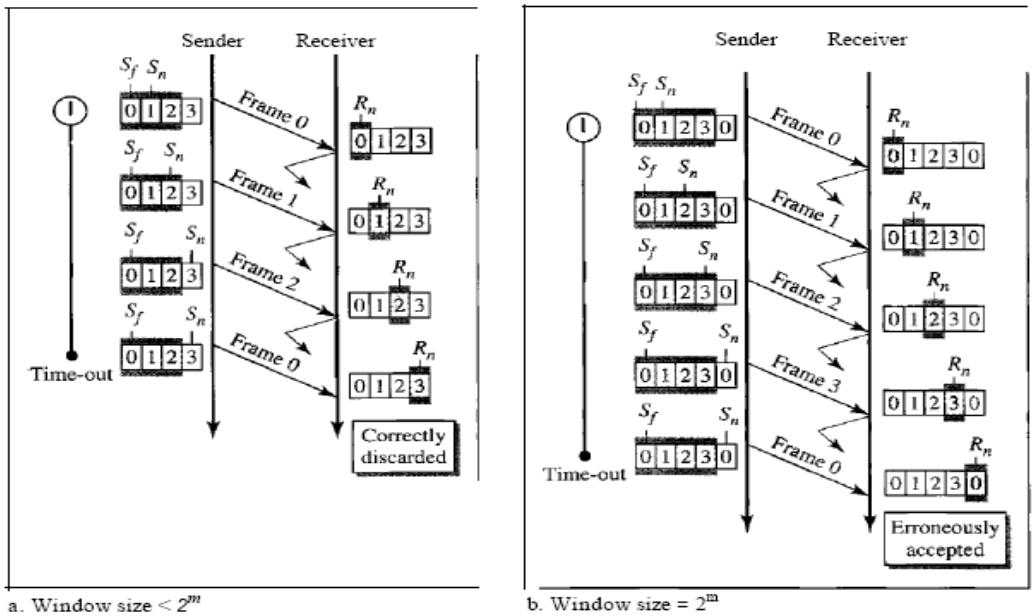

FIGURE 2.14: GO-BACK-N ARQ PROTOCOL FLOW DIAGRAM

- **Selective Repeat Request Protocol[RGPV/Dec 2008/ Dec 2012]**

Design

Multiple frames can be in transit in the forward direction, and multiple acknowledgments in the reverse direction. The idea is similar to Stop-and-Wait ARQ; the difference is that the send window allows us to have as many frames in transition as there are slots in the send window.

Send Window Size

We can now show why the size of the send window must be less than 2m. As an example, we choose m =2, which means the size of the window can be 2m- 1, or 3.

It compares a window size of 3 against a window size of 4. If the size of the window is 3 (less than 22) and all three acknowledgments are lost, the frame timer expires and all three frames are resent. The receiver is now expecting frame 3, not frame 0, so the duplicate frame is correctly discarded. On the other hand, if the size of the window is 4 (equal to 22) and all acknowledgments are lost, the sender will send a duplicate of frame 0. However, this time the window of the receiver expects to receive frame 0, so it accepts frame 0, not as a duplicate, but as the first frame in the next cycle. This is an error.

Windows

The Selective Repeat Protocol also uses two windows: a send window and a receive window.

However, there are differences between the windows in this protocol and the ones in Go-Back-N. First, the size of the send window is much smaller; it is 2m- 1. Second, the receive window is the same size as the send window.

The send window maximum size can be 2m- 1. For example, if m = 4, the sequence numbers go from 0 to 15, but the size of the window is just 8 (it is 15 in the Go-Back-N Protocol). The smaller window size means less efficiency in filling the pipe, but the fact that there are fewer duplicate frames can compensate for this. The protocol uses the same variables as we discussed for Go-Back-N.

The timer for frame starts at the first request, but stops when the ACK for this frame arrives. The timer for frame I start at the second request restarts when a NAK arrives, and finally stops when the last ACK arrives.

The other two timers start when the corresponding frames are sent and stop at the last arrival event.

At the receiver site we need to distinguish between the acceptance of a frame and its delivery to the network layer. At the second arrival, frame 2 arrives and is stored and marked (colored slot), but it cannot be delivered because frame 1 is missing. At the next arrival, frame 3 arrives and is marked and stored, but still none of the frames can be delivered. Only at the last arrival, when finally a copy of frame 1 arrives, can frames 1, 2, and 3 be delivered to the network layer. There are two conditions for the delivery of frames to the network layer: First, a set of consecutive frames must have arrived. Second, the set starts from the beginning of the window.

After the first arrival, there was only one frame and it started from the beginning of the window. After the last arrival, there are three frames and the first one starts from the beginning of the window.

Another important point is that a NAK is sent after the second arrival, but not after the third, although both situations look the same. The reason is that the protocol does not want to crowd the network with unnecessary NAKs and unnecessary resent frames. The second NAK would still be NAKI to inform the sender to resend frame 1 again; this has already been done. The first NAK sent is remembered (using the nakSent variable) and is not sent again until the frame slides. A NAK is sent once for each window position and defines the first slot in the window.

The next point is about the ACKs. Notice that only two ACKs are sent here. The first one acknowledges only the first frame; the second one acknowledges three frames. In Selective Repeat, ACKs are sent when data are delivered to the network layer. If the data belonging to n

frames are delivered in one shot, only one ACK is sent for all of them.

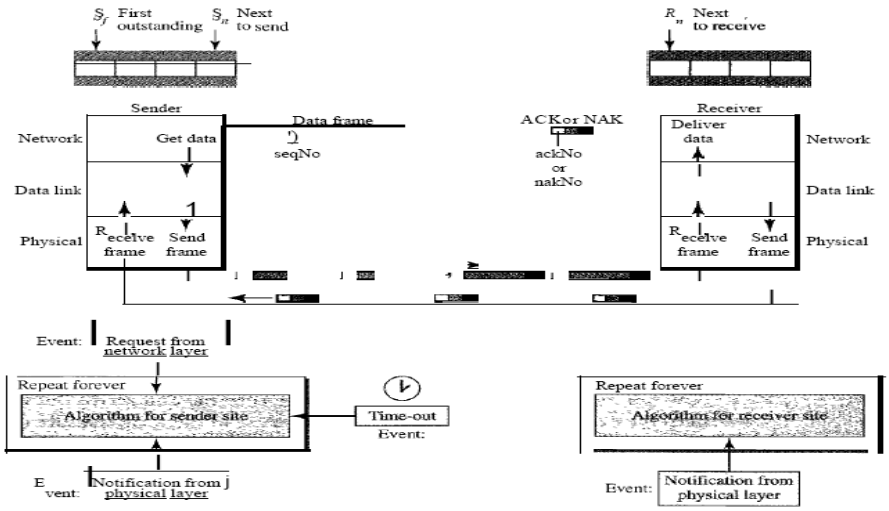

FIGURE 2.15: SELECTIVE REPEAT PROTOCOL

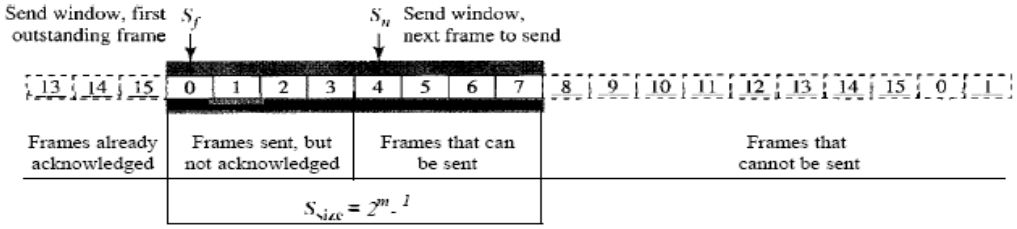

FIGURE 2.16: SENT WINDOW

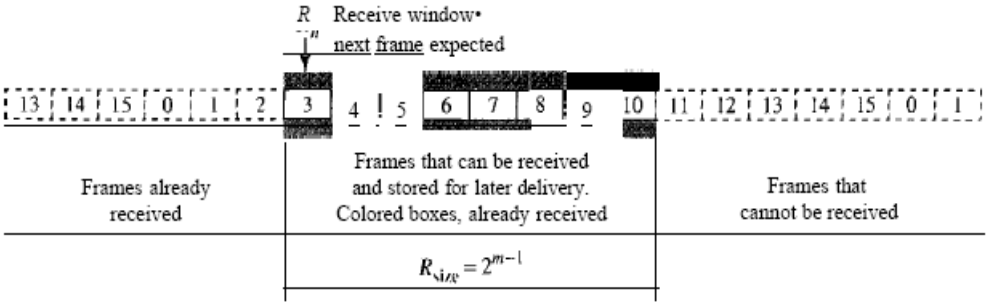

FIGURE 2.17: RECIEVER WINDOW

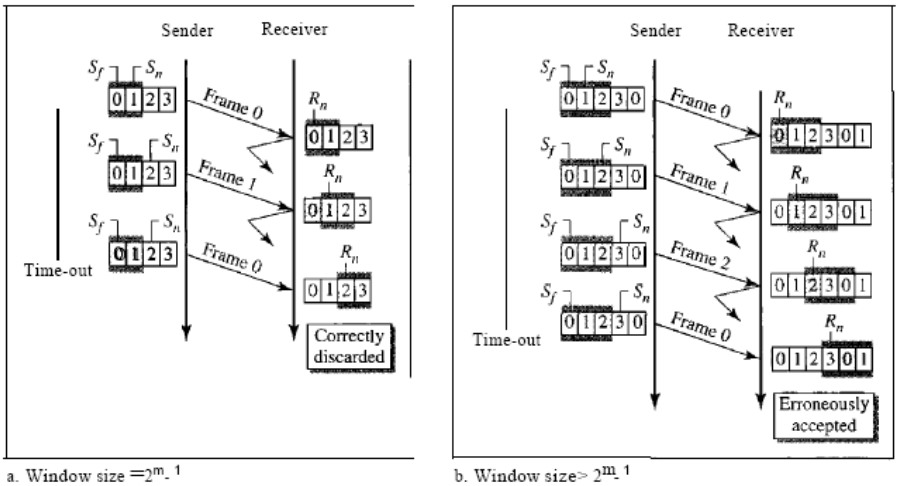

FIGURE 2.18: SELECTIVE REPEAT PROTOCOL FLOW DIAGRAM

Piggybacking[RGPV/Jun 2010, Dec 2012,Jun 2013]

The three protocols are all unidirectional: data frames flow in only one direction although control information such as ACK and NAK frames can travel in the other direction. In real life, data frames are normally flowing in both directions:

From node A to node B and from node B to node A. This means that the control information also needs to flow in both directions. A technique called **piggybacking** is used to improve the efficiency of the bidirectional protocols. When a frame is carrying data from A to B, it can also carry control information about arrived (or lost) frames from B; when a frame is carrying data from B to A, it can also carry control information about the arrived (or lost) frames from A.

S.NO	RGPV QUESTIONS	Year	Marks
Q.1	Explain stop & wait for noisy channel and analyse its performance for channel utilization and optimal frame size.	Dec 2003	7
Q.2	How data link layer- (i) Achieves flow control (ii) Handles duplicates frame at receiver Using stop & wait protocol for noisy channel.	June 2006 June 2007	7
Q.3	Discuss and compare DLL "Go back n" and "Selective repeat" protocols.	June 2004 Dec 2004 Dec 2008	7
Q.4	With the aid of frame sequence diagrams and assuming a selective repeat error control scheme, describe how the following are overcome using both implicit and explicit retransmission- (i) A corrupted information frame (ii) A corrupted ANCK/NAK frame	June 2005	7
Q.5	What is the purpose of timer at the sender site in systems using ARQ? Discuss the size of the Go back n ARQ and selective repeat ARQ sliding window at both the sender sites.	Dec 2010	7
Q.6	In selective repeat protocol, what does the number on a NAK and ACK frame mean?	Dec 2011	7
Q.7	Assuming a send window of X deduce the minimum range of sequence numbers(frame identifiers) required with each of the following error control schemes- (i) A selective repeat (ii) Go back n	Jun 2005	7
Q.8	Explain the working of stop & wait ARQ and Go back N ARQ protocol.	Jun 2013	7
Q.9	Explain the noisy channel protocol: go back N ARQ	Jun 2013	7
Q.10	Explain following: piggybacking	Dec 2012 Jun 2013	7
Q.11	Express why sliding window protocol is superior to stop n wait protocol. Justify your answer with an suitable answer.	Dec 2012	7
Q.12	What is the maximum window size in selective reject ARQ? Justify your answer with an example.	Dec 2012	3.5

Unit-02/Lecture-04

HDLC LAN Protocol

HDLC[RGPV/Jun 2006, Dec 2009,Jun 2010, Dec 2012,Dec 2013, Jun 2014]

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.

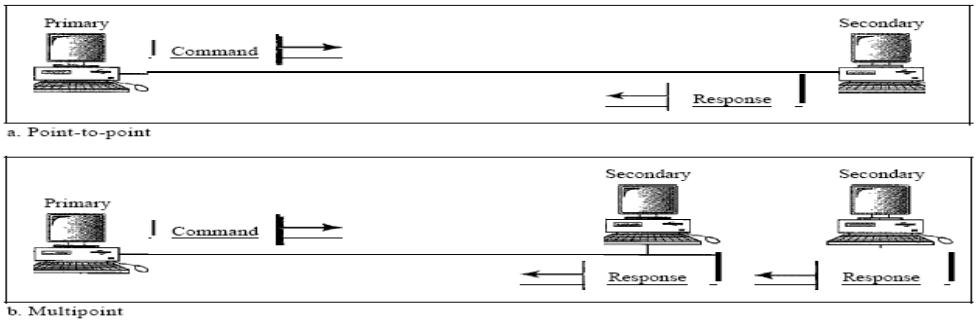

FIGURE 2.19: NRM

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).

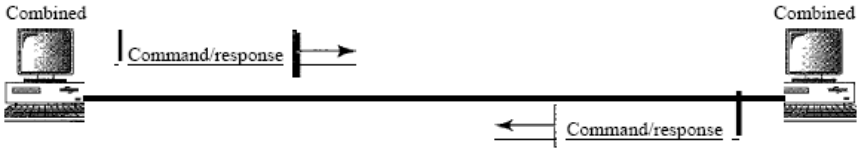

FIGURE 2.20: ABM

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.

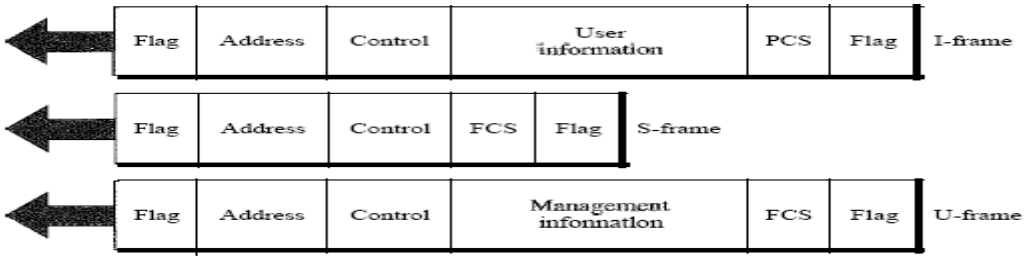

FIGURE 2.21: HDLC FRAMES

DATA LINK CONTROL

Frame Format

Each frame in HDLC may contain up to six fields, 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.

- **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.
- **Address field** It contains the address of the secondary station. If a primary station created the frame, it contains to address. If a secondary creates the frame, it contains 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. 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.
- **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.
- **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.
- **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** The control field determines the type of frame and defines its functionality. 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 PIF bit. The PIP 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).

S.NO	RGPV QUESTIONS	Year	Marks
Q.1	Name the types of HDLC frames and give a brief description of each. In HDLC, what is bit stuffing and why is it needed?	Dec 2010	7
Q.2	Draw the frame format of HDLC protocol. Explain the technique of bit stuffing for data transparency. Explain the use of control, data checksum and address fields of HDLC protocol.	Dec 2009	7
Q.3	Explain about HDLC.	Dec 2013	7
Q.4	What are the transfer modes supported by the HDLC? Describe each.	Dec 2012	7
Q.5	Explain any two of the following data link layer protocols giving characteristics features, frame format and applications: (1) HDLC (2) IEE 802.2 LLC	Jun 2014	14

	(3) PPP		

Unit-02/Lecture-05

Logical link control/ Media access control

IEEE STANDARDS

In 1985, the Computer Society of the IEEE started a project, called Project 802, to set standards to enable intercommunication among equipment from a variety of manufacturers. Project 802 does not seek to replace any part of the OSI or the Internet model. Instead, it is a way of specifying functions of the physical layer and the data link layer of major LAN protocols.

The standard was adopted by the American National Standards Institute (ANSI). In 1987, the International Organization for Standardization (ISO) also approved it as an international standard under the designation ISO 8802.

The IEEE has subdivided the data link layer into two sub layers:

- logical link control (LLC)
- media access control (MAC)

IEEE has also created several physical layer standards for different LAN protocols.

IEEE standard for LANs

- LLC: Logical link control
- MAC: Media access control

the data link layer in the IEEE standard is divided into two sub layers:

- LLC
- MAC

Logical Link Control (LLC) [RGPV/Dec 2009, Jun 2010, Jun 2013, Jun 2014]

Data link control handles framing, flow control, and error control. In IEEE Project 802, flow control, error control, and part of the framing duties are collected into one sublayer called the logical link control. Framing is handled in both the LLC sublayer and the MAC sublayer.

The LLC provides one single data link control protocol for all IEEE LANs. In this way, the LLC is different from the media access control sublayer, which provides different protocols for different LANs. A single LLC protocol can provide interconnectivity between different LANs because it makes the MAC sublayer transparent. One single LLC protocol serving several MAC protocols. Framing LLC defines a protocol data unit (PDU) that is somewhat similar to that of HDLC. The header contains a control field like the one in HDLC; this field is used for flow and error control. The two other header fields define the upper-layer protocol at the source and destination that uses LLC. These fields are called the destination service access point (DSAP) and the source service access point (SSAP). The other fields defined in a typical data link control protocol such as HDLC are moved to the MAC sublayer. In other words, a frame defined in HDLC is divided into a PDU at the LLC sublayer and a frame at the MAC sublayer

Need for LLC The purpose of the LLC is to provide flow and error control for the upper-layer protocols that actually demand these services. For example, if a LAN or several LANs are used in an isolated system, LLC may be needed to provide flow and error control for the application layer protocols. However, most upper-layer protocols media access control that defines the specific access method for each LAN. For example, it defines CSMA/CD as the media access method for Ethernet LANs and the token passing method for Token Ring and Token Bus LANs. Part of the framing function is also handled by the MAC layer.

In contrast to the LLC sublayer, the MAC sublayer contains a number of distinct modules; each defines the access method and the framing format specific to the corresponding LAN protocol.

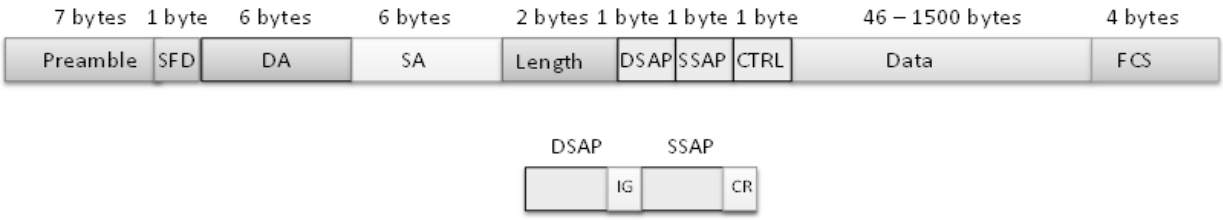

FIGURE 2.22: FRAME FORMAT MAC

Physical Layer

The physical layer is dependent on the implementation and type of physical media used. IEEE defines detailed specifications for each LAN implementation. For example, although there is only one MAC sublayer for Standard Ethernet,

MAC Sublayer [RGPV/Jun 2014]

In Standard Ethernet, the MAC sublayer governs the operation of the access method. It also frames data received from the upper layer and passes them to the physical layer.

Frame Format

The Ethernet frame contains seven fields: preamble, SFD, DA, SA, length or type of protocol data unit (PDU) and upper-layer data. Ethernet does not provide any mechanism for acknowledging received frames, making it what is known as an unreliable medium. Acknowledgments must be implemented at the higher layers.

- **Preamble** The first field of the 802.3 frame contains 7 bytes (56 bits) of alternating 0's and 1's that alerts the receiving system to the coming frame and enables it to synchronize its input timing. The pattern provides only an alert and a timing pulse. The 56-bit pattern allows the stations to miss some bits at the beginning of the frame. The preamble is actually added at the physical layer and is not (formally) part of the frame.
- **Start frame delimiter (SFD)** The second field (1 byte: 10101011) signals the beginning of the frame. The SFD warns the station or stations that this is the last chance for synchronization. The last 2 bits is 11 and alerts the receiver that the next field is the destination address.
- **Destinations address (DA)** The DA field is 6 bytes and contains the physical address of the destination station or stations to receive the packet.
- **Sources address (SA)** The SA field is also 6 bytes and contains the physical address of the sender of the packet.
- **Length or type** This field is defined as a type field or length field. The original Ethernet used this field as the type field to define the upper-layer protocol using the MAC frame. The IEEE standard used it as the length field to define the number of bytes in the data field. Both uses are common today.
- **Data** This field carries data encapsulated from the upper-layer protocols. It is a minimum of 46 and a maximum of 1500 bytes CRC. The last field contains error detection information, in this case a CRC-32
- **Frame Length** Ethernet has imposed restrictions on both the minimum and maximum lengths of a frame,
  Minimum payload length: 46 bytes
  Maximum payload length: 1500 bytes-1

  Minimum frame length: 512 bits or 64 bytes
  Maximum frame length. 12,144 bits or 1518 bytes

  An Ethernet frame needs to have a minimum length of 512 bits or 64 bytes. Part of this length is the header and the trailer. If we count 18 bytes of header and trailer (6 bytes of source address, 6 bytes of destination address, 2 bytes of length or type, and 4 bytes of

CRC), then the minimum length of data from the upper layer is 64 - 18 = 46 bytes. If the upper-layer packet is less than 46 bytes, padding is added to make up the difference.
The standard defines the maximum length of a frame (without preamble and SFD field) as 1518 bytes. If we subtract the 18 bytes of header and trailer, the maximum length of the payload is 1500 bytes. The maximum length restriction has two historical reasons. First, memory was very expensive when Ethernet was designed: a maximum length restriction helped to reduce the size of the buffer. Second, the maximum length restriction prevents one station from monopolizing the shared medium, blocking other stations that have data to send.

- **Frame length:**
  Minimum: 64 bytes (512 bits) Maximum: 1518 bytes (12,144 bits)

- **Addressing**
  Each station on an Ethernet network (such as a PC, workstation, or printer) has its own network interface card (NIC). The NIC fits inside the station and provides the station with a 6-byte physical address. As shown in Figure 13.6, the Ethernet address is 6 bytes (48 bits), normally written in hexadecimal notation, with a colon between the bytes.
- Unicast
- Multicast
- Broadcast Addresses

A source address is always a unicast address-the frame comes from only one station. The destination address, however, can be unicast, multicast, or broadcast.
If the least significant bit of the first byte in a destination address is 0, the address is unicast; otherwise, it is multicast.
If the bit is 0, the address is unicast; otherwise, it is multicast.
A unicast destination address defines only one recipient; the relationship between the sender and the receiver is one-to-one. A multicast destination address defines a group of addresses; the relationship between the sender and the receivers is one-to-many. The broadcast address is a special case of the multicast address; the recipients are all the stations on the LAN. A broadcast destination address is forty-eight 1's. The broadcast destination address is a special case of the multicast address in which all bits are 1's.

S.NO	RGPV QUESTIONS	Year	Marks
Q.1	Explain the concept of Ethernet frame and explain the meaning of each field in the frame?	Jun 2014	7
Q.2	Explain the following: IEEE 802.2 LLC frame format	Dec 2009 Jun 2010 Jun 2013	7
Q.3	Explain any two of the following data link layer protocols giving characteristics features, frame format and applications: (1) HDLC (2) IEE 802.2 LLC (3) PPP	Jun 2014	14
Q.4	Explain IEE 802.3/ ether net frame format justifying utility of each field in the frame.	Jun 2014	3.5

UNIT 2/LECTURE 06

SERIAL LINE INTERNET PROTOCOL (SLIP)

Serial Line Internet Protocol (SLIP)

SLIP Basic Data Framing Method and General Operation

An IP datagram is passed down to SLIP, which breaks it into bytes and sends them one at a time over the link. After the last byte of the datagram, a special byte value is sent that tells the receiving device that the datagram has ended. This is called the SLIP END character, and has a byte value of 192 decimal (C0 hexadecimal, 11000000 binary). And that's basically it: take the whole datagram, send it one byte at a time, and then send the byte 192 to delimit the end of the datagram.

A minor enhancement to this basic operation is to **precede** the datagram by an END character as well. The benefit of this is that it clearly separates the start of the datagram from anything that preceded it. To see why this might be needed, suppose at a particular time we have only one datagram to send, datagram #1. So, we send #1, and then send the END character to delimit it. Now, suppose there is a pause before the next datagram shows up. During that time we aren't transmitting, but if there is line noise, the other device might pick up spurious bytes here and there. If we later receive datagram #2 and just start sending it, the receiving device might think the noise bytes were part of datagram #2.

Starting datagram #2 off with an END character tells the recipient that anything received between this END character and the previous one is a separate datagram. If that's just noise, then this “noise datagram” is just gibberish that will be rejected at the IP layer. Meanwhile, it doesn't corrupt the real datagram we wish to send. If no noise occurred on the line between datagrams then the recipient will just see the END at the start of datagram #2 right after the one at the end of #1, and will ignore the “null datagram” between the two.

Escaping Special Characters

There is only one other issue SLIP deals with. If the END character is 192 decimal, what happens if the byte value 192 appears in the datagram itself? Transmitting it “as is” would fool the recipient into thinking the datagram ended prematurely. To avoid this, a special Escape character (ESC) is defined, which has a decimal value of 219 (DB in hex, 11011011 in binary). The term “escape” means that this symbol conveys the meaning “this byte and the next one have a special meaning”. When a value of 192 appears in the datagram, the sending device replaces it by the ESC character (219 decimal) followed by the value 220 decimal. Thus, a single “192” becomes “219 220” (or “DB DC” in hexadecimal). The recipient translates back from “219 220” to “192”.

This leaves one final problem: what happens if the **escape character itself** is in the original datagram? That is, what if there's a byte value of 219 in the IP datagram to be sent? This is handled by a similar substitution: instead of “219” we put “219 221”.

So in summary, this is basically everything SLIP does:

- Break an IP datagram into bytes.
- Send the END character (value “192”) after the last byte of the datagram; in better implementations, send the END character before the first byte as well.
- If any byte to be sent in the datagram is “192”, replace it with “219 220”.
- If any byte to be sent is “219”, replace it with “219 221”.

Figure shows an example of how SLIP works, including the escaping of special characters, using a mock IP datagram.

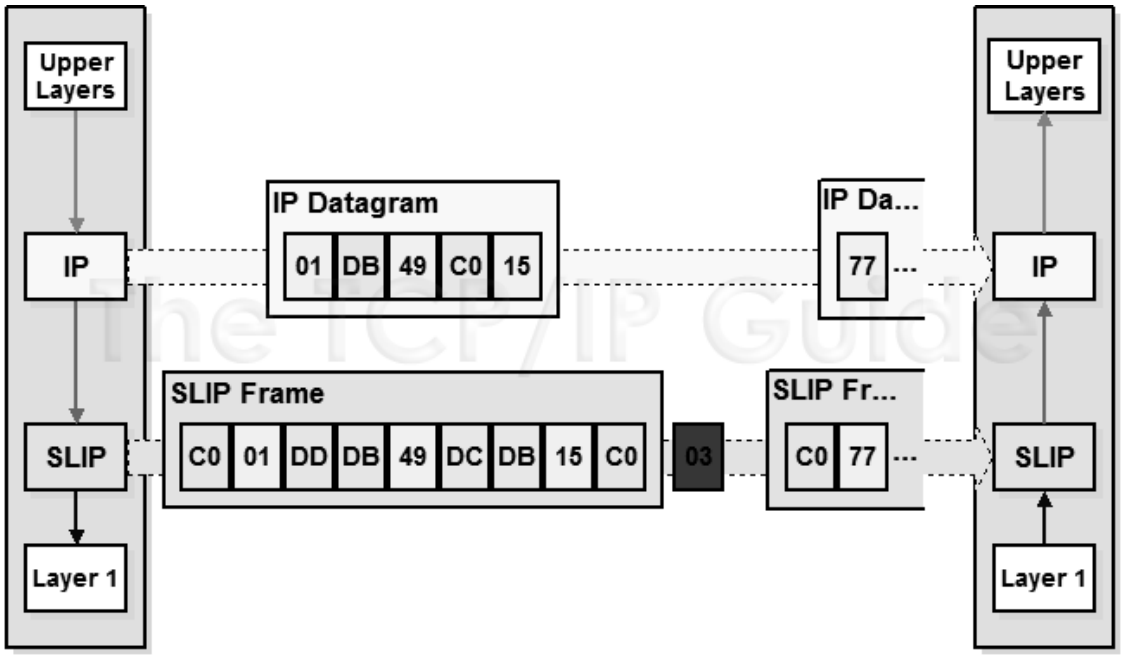

FIGURE 2.23: OPERATION OF THE SERIAL LINE INTERNET PROTOCOL (SLIP)

IP datagrams are passed down to the SLIP software at layer two (a simplified one with only five bytes is shown here). There, they are framed by surrounding them with END characters (hexadecimal value C0h, shown in orange). Special characters with hexadecimal values DBh and C0h are replaced by two-byte sequences. Note that the presence of the bracketing END characters forces the receiving device to see the noise byte (03h, in red) as a separate IP datagram, rather than part of either of the real ones. It will be rejected when passed up to the IP layer.

S.NO	RGPV QUESTIONS	Year	Marks
Q.1	Explain the SLIP	June 2010	7

Unit-02/Lecture-07

POINT-TO-POINT PROTOCOL

POINT-TO-POINT PROTOCOL[RGPV/[Dec 2004, Dec 2009, Jun 2010, Jun 2014]

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). Today, millions of Internet users who need to connect their home computers to the server of an Internet service provider use PPP. The majority of these users have a traditional modem; they are connected to the Internet through a telephone line, which provides the services of the physical layer, manage the transfer of data; there is a need for a point-to-point protocol at the data link layer.

PPP provides several services:

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

On the other hand, to keep PPP simple, several services are missing:

- PPP does not provide flow control. A sender can send several frames one after another with no concern about overwhelming the receiver.
- 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.
- PPP does not provide a sophisticated addressing mechanism to handle frames in a multipoint configuration.

Framing

PPP is a byte-oriented protocol.

Frame Format

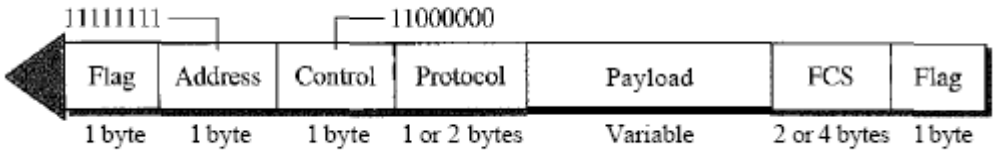

FIGURE 2.24: FRAME FORMAT PPP

- **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
- **Address** The address field in this protocol is a constant value and set to 11111111 (broadcast address). During negotiation, the two parties may agree to omit this byte.
- **Control** This field is set to the constant value 11000000 (imitating unnumbered frames in HDLC). 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,

we dont take any liability for the notes correctness. http://www.rgpvonline.com

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. This field is by default 2 bytes long, but the two parties can agree to use only I byte.
- **Payload field** This field carries either the user data or other information .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 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 flag like 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

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

- Failed
- Carrier
- Dropped
- Terminate
- Done
- Carrier
- Detected
- Failed

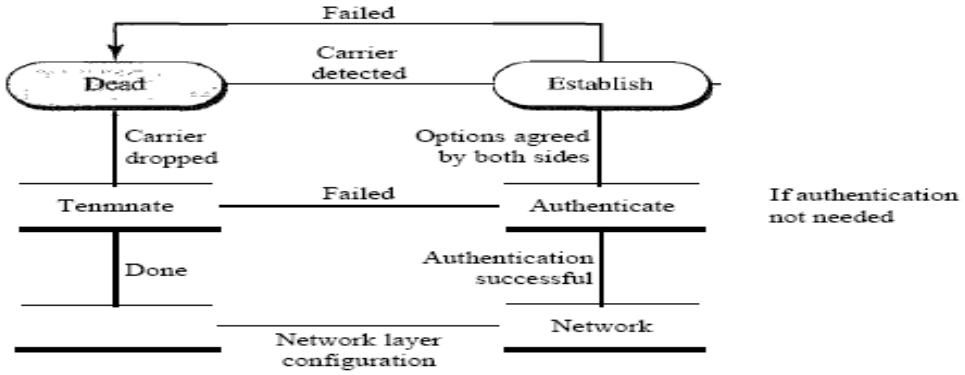

FIGURE 2.25: TRANSITION PHASE PPP

- **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 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, If the result is successful, the connection goes to the networking phase; otherwise, it goes to the termination phase.

we dont take any liability for the notes correctness. http://www.rgpvonline.com

- **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.

S.NO	RGPV QUESTIONS	Year	Marks
Q.1	What is PPP?	Jun 2010	7
Q.2	Write a brief note on point to point protocol.	Dec 2004 Dec 2009	7
Q.3	Explain any two of the following data link layer protocols giving characteristics features, frame format and applications: (1) HDLC (2) IEE 802.2 LLC (3) PPP	Jun 2014	14

UNIT 2/LECTURE 08/ADDITIONAL TOPIC

Project 802

IEEE802 refers to a family of IEEE standards dealing with local area networks and metropolitan area networks.

More specifically, the IEEE 802 standards are restricted to networks carrying variable-size packets. (By contrast, in cell relay networks data is transmitted in short, uniformly sized units called cells. Isochronous networks, where data is transmitted as a steady stream of octets, or groups of octets, at regular time intervals, are also out of the scope of this standard.) The number 802 was simply the next free number IEEE could assign, though "802" is sometimes associated with the date the first meeting was held — February 1980.

The services and protocols specified in IEEE 802 map to the lower two layers (Data Link and Physical) of the seven-layer OSI networking reference model. In fact, IEEE 802 splits the OSI Data Link Layer into two sub-layers named Logical Link Control (LLC) and Media Access Control (MAC), so that the layers can be listed like this:

- Data link layer
  - LLC Sublayer
  - MAC Sublayer
- Physical layer

The IEEE 802 family of standards is maintained by the IEEE 802 LAN/MAN Standards Committee (LMSC). The most widely used standards are for the Ethernet family, Token Ring, Wireless LAN, Bridging and Virtual Bridged LANs. An individual Working Group provides the focus for each area.

Working groups[edit]

Name	Description	Note
IEEE 802.1	Bridging (networking) and Network Management	
IEEE 802.2	LLC	inactive
IEEE 802.3	Ethernet	
IEEE 802.4	Token bus	disbanded
IEEE 802.5	Defines the MAC layer for a Token Ring	inactive
IEEE 802.6	MANs (DQDB)	disbanded
IEEE 802.7	Broadband LAN using Coaxial Cable	disbanded
IEEE 802.8	Fiber Optic TAG	disbanded
IEEE 802.9	Integrated Services LAN (ISLAN or isoEthernet)	disbanded
IEEE 802.10	Interoperable LAN Security	disbanded
IEEE 802.11	Wireless LAN (WLAN) & Mesh (Wi-Fi certification)	
IEEE 802.12	100BaseVG	disbanded

IEEE 802.13	Unused	Reserved for Fast Ethernet development
IEEE 802.14	Cable modems	disbanded
IEEE 802.15	Wireless PAN	
IEEE 802.15.1	Bluetooth certification	
IEEE 802.15.2	IEEE 802.15 and IEEE 802.11 coexistence	
IEEE 802.15.3	High-Rate wireless PAN (e.g., UWB, etc.)	
IEEE 802.15.4	Low-Rate wireless PAN (e.g., ZigBee, WirelessHART, MiWi, etc.)	
IEEE 802.15.5	Mesh networking for WPAN	
IEEE 802.15.6	Body area network	
IEEE 802.16	Broadband Wireless Access (WiMAX certification)	
IEEE 802.16.1	Local Multipoint Distribution Service	
IEEE 802.17	Resilient packet ring	
IEEE 802.18	Radio Regulatory TAG	
IEEE 802.19	Coexistence TAG	
IEEE 802.20	Mobile Broadband Wireless Access	
IEEE 802.21	Media Independent Handoff	
IEEE 802.22	Wireless Regional Area Network	
IEEE 802.23	Emergency Services Working Group	
IEEE 802.24	Smart Grid TAG	New (November, 2012)
IEEE 802.25	Omni-Range Area Network	Not yet ratified

we dont take any liability for the notes correctness. http://www.rgpvonline.com

ADDRESSES Four levels of addresses are used in an internet employing the TCP/IP protocols:

- physical (link) addresses
- logical (IP) addresses
- port addresses
- specific addresses

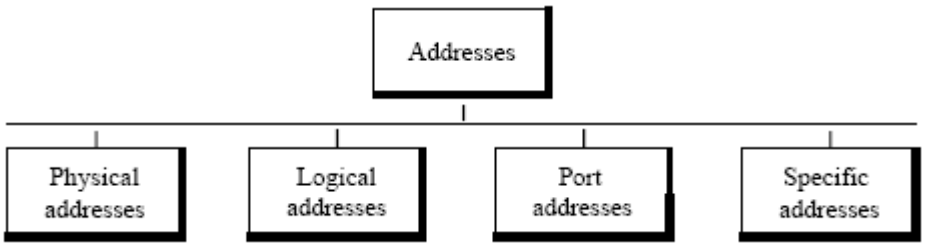

FIGURE 2.26: ADDRESS

- **Physical Addresses** The physical address, also known as the link address, is the address of a node as defined by its LAN or WAN. It is included in the frame used by the data link layer. It is the lowest-level address.
  The physical addresses have authority over the network (LAN or WAN). The size and format of these addresses vary depending on the network.

- **Logical Addresses** Logical addresses are necessary for universal communications that are independent of underlying physical networks. Physical addresses are not adequate in an internetwork environment where different networks can have different address formats. A universal addressing system is needed in which each host can be identified uniquely, regardless of the underlying physical network. The logical addresses are designed for this purpose. A logical address in the Internet is currently a 32-bit address that can uniquely define a host connected to the Internet. No two publicly addressed and visible hosts on the Internet can have the same IP address.

- **Port Addresses** The IP address and the physical address are necessary for a quantity of data to travel from a source to the destination host. However, arrival at the destination host is not the final objective of data communications on the Internet. A system that sends nothing but data from one computer to another is not complete. Today, computers are devices that can run multiple processes at the same time. The end objective of Internet communication is a process communicating with another process. For example, computer A can communicate with computer C by using TELNET. At the same time, computer A communicates with computer B by using the File Transfer Protocol (FTP). For these processes to receive data simultaneously, we need a method to label the different processes.
  In other words, they need addresses. In the TCP/IP architecture, the label assigned to a process is called a port address. A port address in TCP/IP is 16 bits in length.

- **Specific Addresses**
  Some applications have user-friendly addresses that are designed for that specific address. Examples include the e-mail address and the Universal Resource Locator (URL).The first defines the recipient of an e-mail, the second is used to find a document on the World Wide Web. These addresses, however, get changed to the corresponding port and logical addresses by the sending computer.