SYLLABUS

Unit 1: Hrs: 06
Layered Tasks, OSI model, Layers in OSI model, TCP/IP Suite, Addressing.
Telephone and cable networks for data transmission, Telephone Networks, Dial up modem, DSL, Cable TV for data transmission.

Unit 2: Data Link control Hrs: 07
Framing, Flow & Error control, Protocols, Noiseless channels & Noisy channels, HDLC.

Unit 3: Multiple Accesses Hrs: 06
Random access, Controlled access, channelization

Unit 4: Hrs: 07
IEEE standards, standard Ethernet, changes in the standards, Fast Ethernet, Gigabit Ethernet, Wireless LAN IEEE 802.11

Unit 5: Hrs: 06
Connecting LANs, Backbone and virtual LANs, Connecting devices, Backbone networks, Virtual LANs.

Unit 6: Hrs: 07
Network layer, Logical addressing, IPv4 addresses, IPv6 addresses, IPv4 and IPv6 transition from IPv4 to IPv6.

Unit 7: Hrs: 06
Delivery, Forwarding, Unicast Routing protocols, Multicast Routing protocols

Unit 8: Hrs: 06
Transport layer process to process delivery, UDP, TCP, Domain name system, Resolution.

Prescribed & Reference Books:

Sl. No.	Particulars	Book Title	Book Author	Book Publications
1	Prescribed Books	Data Communication & Networking	B Forouzan	4_{th} Ed, 2006, TMH
2	Reference Books	Computer Networks	James F. Kursoe, Keith W. Ross	2_{nd} Ed, 2003, Pearson
3		Introduction to Data communication & Networking	Wayne Tomasi	2007, Pearson

DATA COMMUNICATIONS INTRODUCTION

Data communications are the exchange of data between two devices via some form of transmission medium such as a wire cable. For data communications, to occur the communicating devices must be part of a communication system made up of a combination of hardware (physical equipment) and software (programs). The effectiveness of a data communications system depends on four fundamental characteristics: **delivery, accuracy, timeliness,** and **jitter.**

* **Delivery**: The system must deliver data to the correct destination. Data must be received by the intended device or user and only by that device or user.
* **Accuracy:** The system must deliver the data accurately. Data that have been altered in transmission and left uncorrected are unusable.
* **Timeliness:** The system must deliver data in a timely manner. Data delivered late are useless. In the case of video and audio, timely delivery means delivering data as they are produced, in the same order that they are produced, and without significant delay. This kind of delivery is called real-time transmission.
* **Jitter:** Jitter refers to the variation in the packet arrival time. It is the uneven delay in the delivery of audio or video packets.

COMPONENTS

A data communications system has five components:

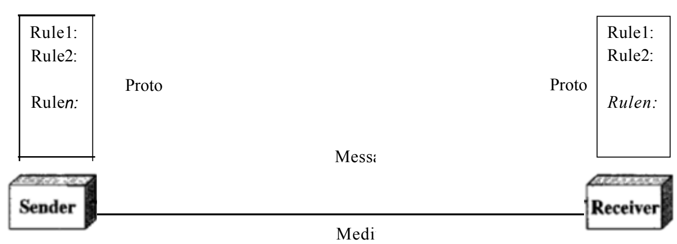


Figure 1.1

Message: The message is the information (data) to be communicated. Popular forms of information include text, numbers, pictures, audio, and video.

Sender: The sender is the device that sends the data message. It can be a computer, workstation, telephone handset, video camera, and so on.

Receiver. The receiver is the device that receives the message. It can be a computer, workstation, telephone handset, television, and so on.

Transmission medium: The transmission medium is the physical path by which a message travels from sender to receiver. Some examples of transmission media include twisted-pair wire, coaxial cable, fiber-optic cable, and radio waves.

Protocol: A protocol is a set of rules that govern data communications. It represents an agreement between the communicating devices.

DATA REPRESENTATION:

Information today comes in different forms such as text, numbers, images, audio, and video.

Text: In data communications, text is represented as a bit pattern, a sequence of bits (Os or Is). Different sets of bit patterns have been designed to represent text symbols. Each set is called a code, and the process of representing symbols is called coding.

Numbers: Numbers are also represented by bit patterns. However, a code such as ASCII is not used to represent numbers; the number is directly converted to a binary number to simplify mathematical operations.

Images: Images are also represented by bit patterns. In its simplest form, an image is composed of a matrix of pixels (picture elements), where each pixel is a small dot. The size of the pixel depends on the resolution

Audio: Audio refers to the recording or broadcasting of sound or music. Audio is by nature different from text, numbers, or images. It is continuous, not discrete.

Video: Video refers to the recording or broadcasting of a picture or movie. Video can either be produced as a continuous entity (e.g., by a TV camera), or it can be a combination of images, each a discrete entity, arranged to convey the idea of motion.

Unit 1:

Network Architecture

Syllabus:

Section 1: Layered Tasks, OSI model, Layers in OSI model, TCP/IP Suite, Addressing
Section 2: Telephone and cable networks for data transmission, Telephone Networks, Dial up modem, DSL, Cable TV for data transmission.

Hrs: 06

Recommended readings:

Text Book: Data Communication & Networking, B Forouzan, 4e, TMH

Sec 1: Chapter 2 – Page 27 to 50
Sec 2: Chapter 9 – Page 241 to 260

1.1 Layered Tasks:

Consider two friends who communicate through mail

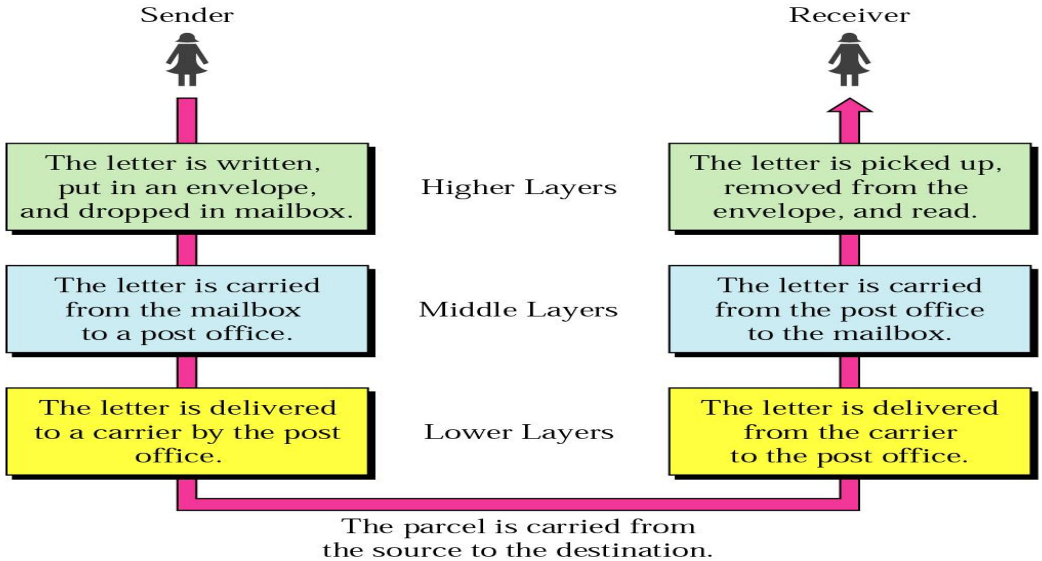


Figure1.2: Tasks involved in sending a letter

In the above figure 1.2, consists of a sender, a receiver & a carrier that transports the letter. There is a hierarchy of tasks.

At the sender site:

- The sender writes the letter, inserts the letter in an envelope, writes the sender & receiver addresses, and drops the letter in a mail box.
- Middle layer:_The letter is picked up by a letter carrier and delivered to the post office.
- Lower Layer:_The letter is sorted at the post office, a carrier transports the letter.

On the way:

The letter is then on its way to the recipient. On the way to the recipient's local post office, the letter may actually go through a central office. In addition, it may be transported by truck, train, airplane, boat, or a combination of these.

At the receiver site:

- Lower layer:_The carrier transports the letter to the post office.
- Middle Layer:_The letter is sorted & delivered to the recipient's mailbox.
- Higher Layer:_The receiver picks up the letter, opens the envelope, and reads it.

Hierarchy:

The task of transporting the letter between the sender and the receiver is done by the carrier. The tasks must be done in the order given in the hierarchy. At the sender site, the letter must be written and dropped in the mailbox before being picked up by the letter carrier and delivered to the post office. At the receiver site, the letter must be dropped in the recipient mailbox before being picked up & read by the recipient.

Services:

Each layer at the sending site uses the services of the layer immediately below it. The sender at the higher layer uses the services of the middle layer. The middle layer uses the services of the lower layer. The lower layer uses the services of the carrier.

1.2 The OSI model:

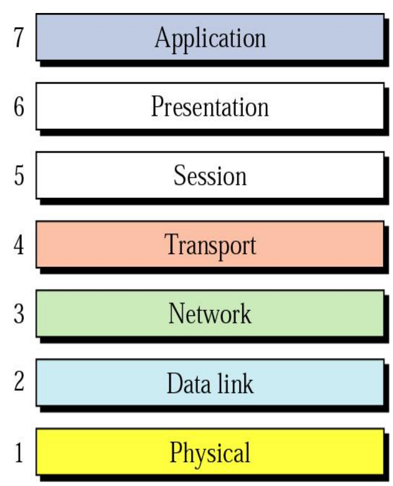


Figure 1.2 Seven layers of the OSI model

The OSI model shown in figure 1.2 is based on the proposal developed by the International Standards Organization (ISO) as a first step towards international standardization of the protocols used in the various layers. The model is called the OSI **(Open System Interconnection)** reference model because it deals with connecting open systems, i.e., systems that are open for communication with other systems. The purpose of the OSI model is to show how to facilitate communication between different systems without requiring changes to the logic of the underlying hardware and software.

The OSI model is not a protocol; it is a model for understanding and designing a network architecture that is flexible, robust and interoperable. The OSI model is a layered framework for the design of network systems that allows communication between all types of computer systems. It consists of seven separate but related layers, each of which defines a part of the process of moving information across a network. The principles that were applied to arrive at the seven layers are as follows:

* A layer should be created where a different level of abstraction is needed.

* Each layer should perform a well-defined function.
* The function of each layer should be chosen with an eye toward defining internationally standardized protocols.
* The layer boundaries should be chosen to minimize the information flow across the interfaces.
* The number of layers should be large enough that distinct functions need not be thrown together in the same layer out of necessity and small enough that the architecture does not become unwieldy.

Layered Architecture:

✓ The OSI model is composed of seven layers: Physical, Data link, Network, Transport, Session, Presentation, Application layers. Figure1.3 shows the layers involved when a message travels from A to B, it may pass through many intermediate nodes. These intermediate nodes involve only the first 3 layers of the OSI model.

✓ Within a single machine, each layer calls upon the services of the layer just below it, layer 3 for example. Uses the services provided by layer 2 & provides services for layer 4. Between machines, layer X on one machine communicates with layer X on another machine. This communication is governed by an agreed upon series of rules & Conventions called protocols.

✓ The processes on each machine that communicate at a given layer are called peer – to – peer processes. Communication between machines is therefore a peer – to –peer process using the protocols appropriate to a given layer.

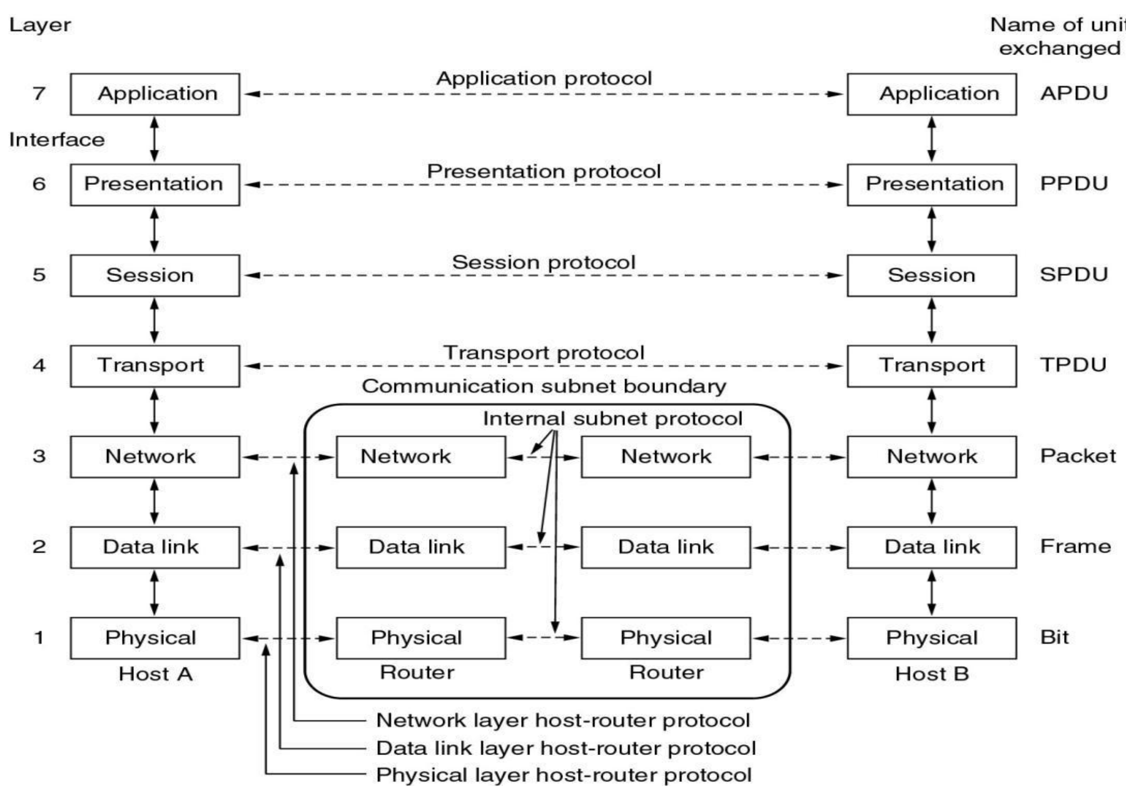
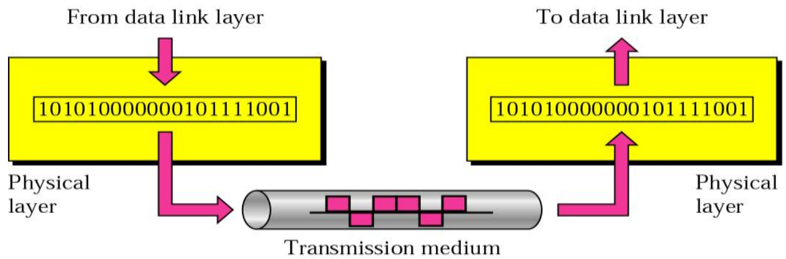


Fig (iii): Interaction between layers in the OSI model

1.3 Layers in the OSI model:

* **Physical Layer:**

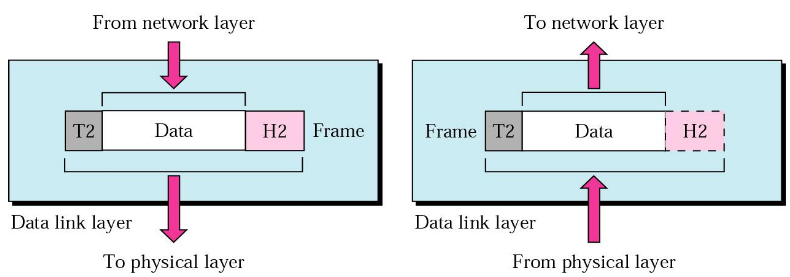
Physical Layer is responsible for movements of individual bits from one node to the next



- Physical characteristics of interfaces & medium, type of transmission medium.
- Representation of bits.
- Data rate.
- Synchronization of bits.
- Line configuration.
- Physical topology – Mesh, Star, Ring, Bus, Hybrid.
- Transmission mode – Simplex, Half-duplex, Full-duplex.

ii) Data Link Layer:

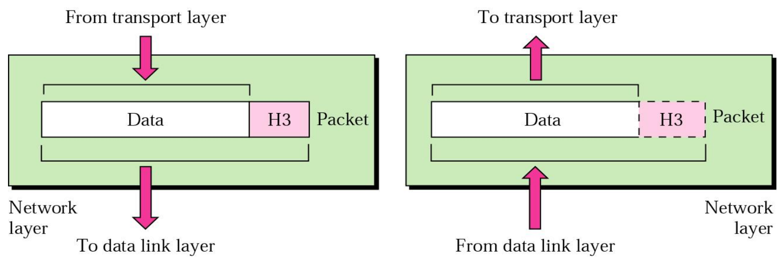
Data link layer is responsible for moving frames from one node to the next.



- Framing.
- Physical addressing.
- Flow control.
- Error control.
- Access control.

iii) Network Layer:

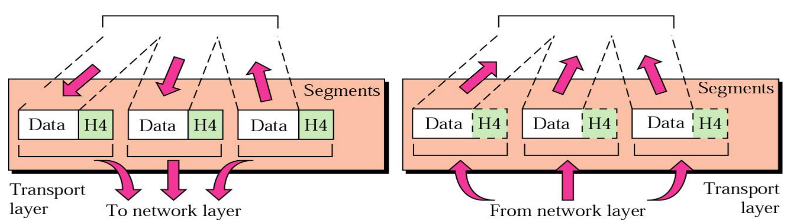
Network layer is responsible for the delivery of individual packets from the source host to the destination host.



- Logical addressing.
- Routing.

iv) Transport Layer:

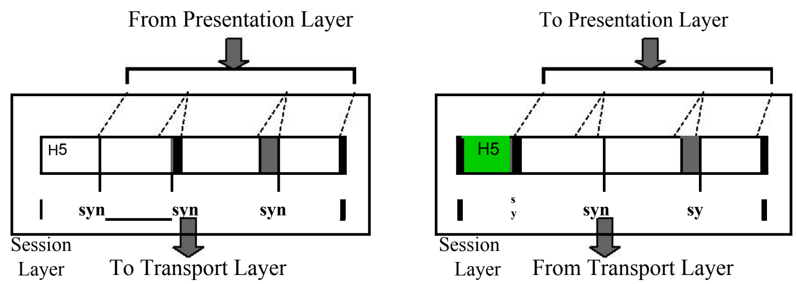
Transport layer is responsible for the delivery of a message from one process to another.



- Service-point addressing.
- Segmentation and reassembly.
- Connection control.
- Flow control.
- Error control.

v) Session Layer:

The session layer is responsible for dialog control and synchronization.



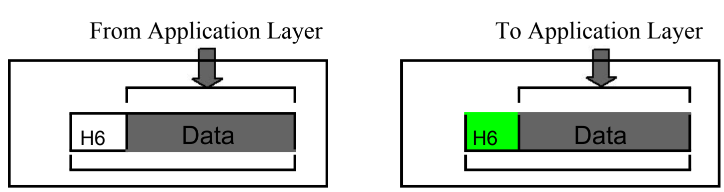
- Dialog control.
- Synchronization.

Dialog control: The session layer allows two systems to enter into a dialog. It allows the communication between two processes to take place in either half duplex(one way at a time) or full-duplex (two ways at a time) mode.

Synchronization. :The session layer allows a process to add checkpoints, or synchronization points, to a stream of data. For example, if a system is sending a file of 2000 pages, it is advisable to insert checkpoints after every 100 pages to ensure that each 100-page unit is received and acknowledged independently. In this case, if a crash happens during the transmission of page 523, the only pages that need to be resent after system recovery are pages 501 to 523. Pages previous to 501 need not be resent. Figure illustrates the relationship of the session layer to the transport and presentation layers

vi) Presentation Layer:

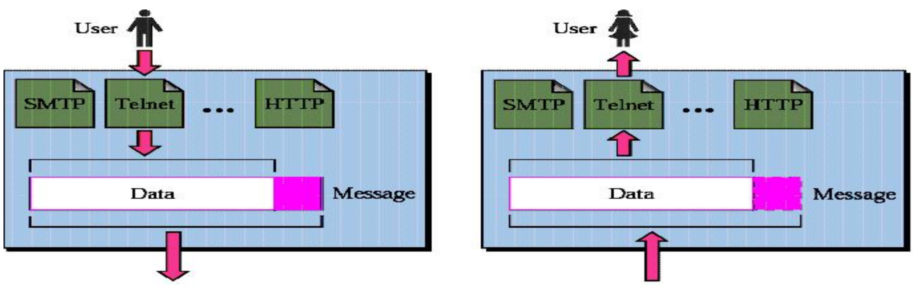
The Presentation layer is responsible for translation, compression, and encryption.



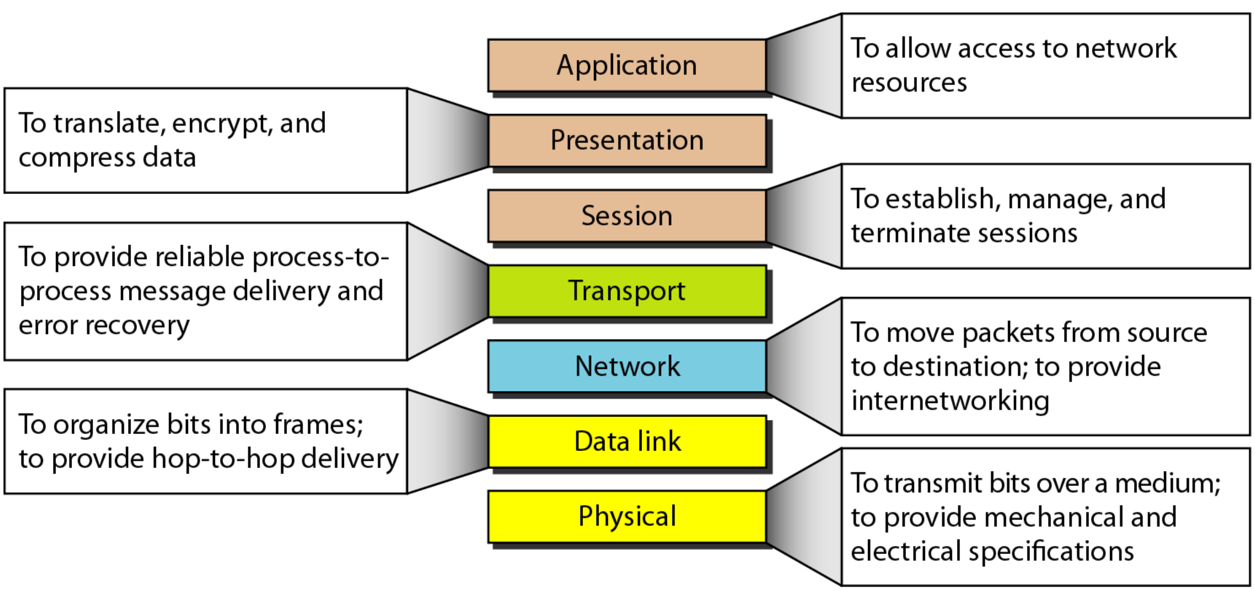
- Concerned with syntax and semantics of the information.
- Translation.
- Encryption.
- Compression.

vii) Application Layer:

The Application layer is responsible for providing services to the user.



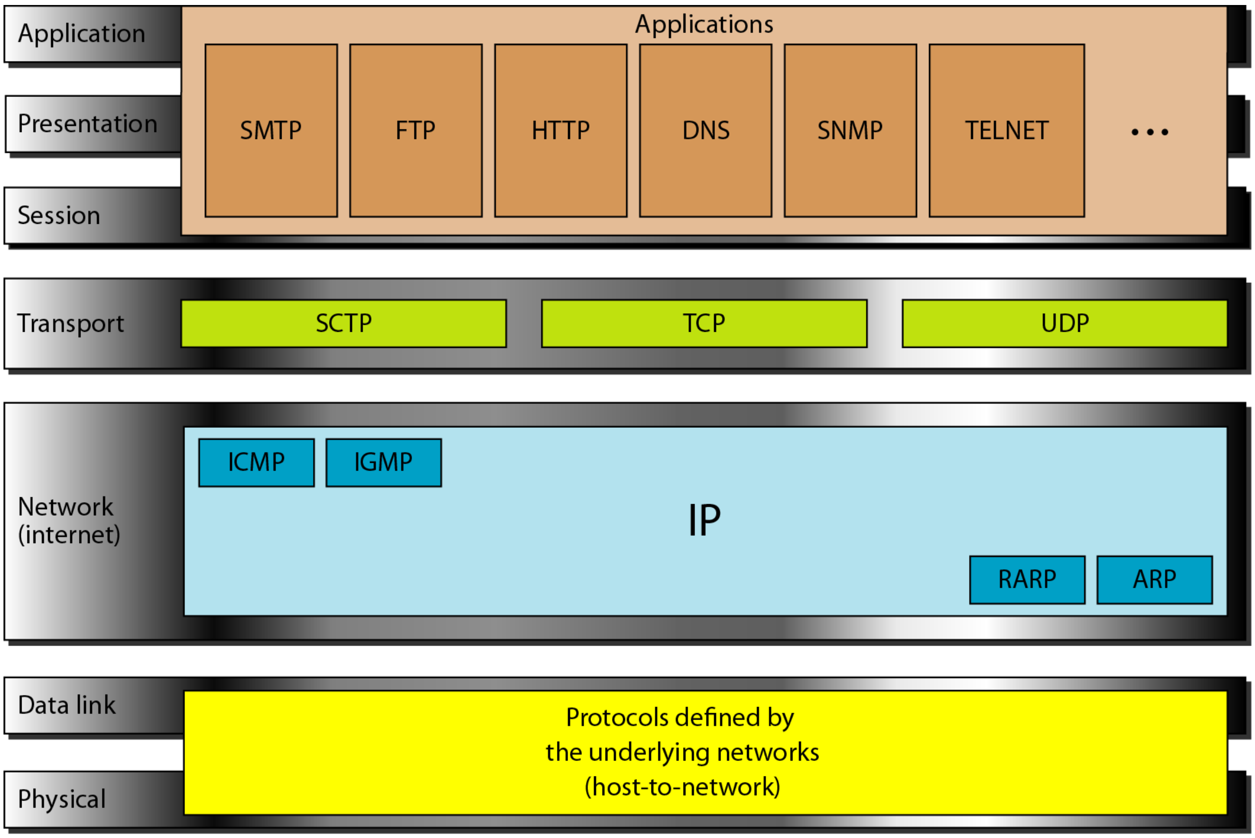
- Network virtual terminal.
- File transfer, access, and management.
- Mail services.
- Directory Services



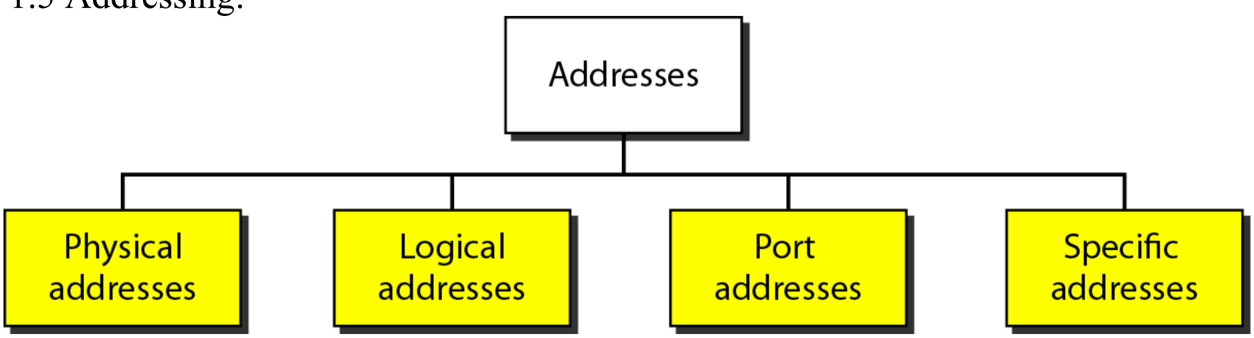
OSI Model			
	Data unit	**Layer**	**Function**
Host layers	Data	7. Application	Network process to application
		6. Presentation	Data representation and encryption
		5. Session	Interhost communication
	Segment	4. Transport	End-to-end connections and reliability
Media layers	Packet	3. Network	Path determination and logical addressing
	Frame	2. Data Link	Physical addressing
	Bit	1. Physical	Media, signal and binary transmission

1.3TCP/IP Protocol suite:

The TCP/IP protocol suite has four layers: Host – to – Network, Internet, Transport and Application. Comparing TCP/IP to OSI model: the Host – to – Network layer is equivalent to the combination of physical and data link layers, the Internet layer is equivalent to the network layer, the Transport layer in TCP/IP taking care of part of the duties of the session layer, and the application layer is roughly doing the job of the session, presentation, & application layers.

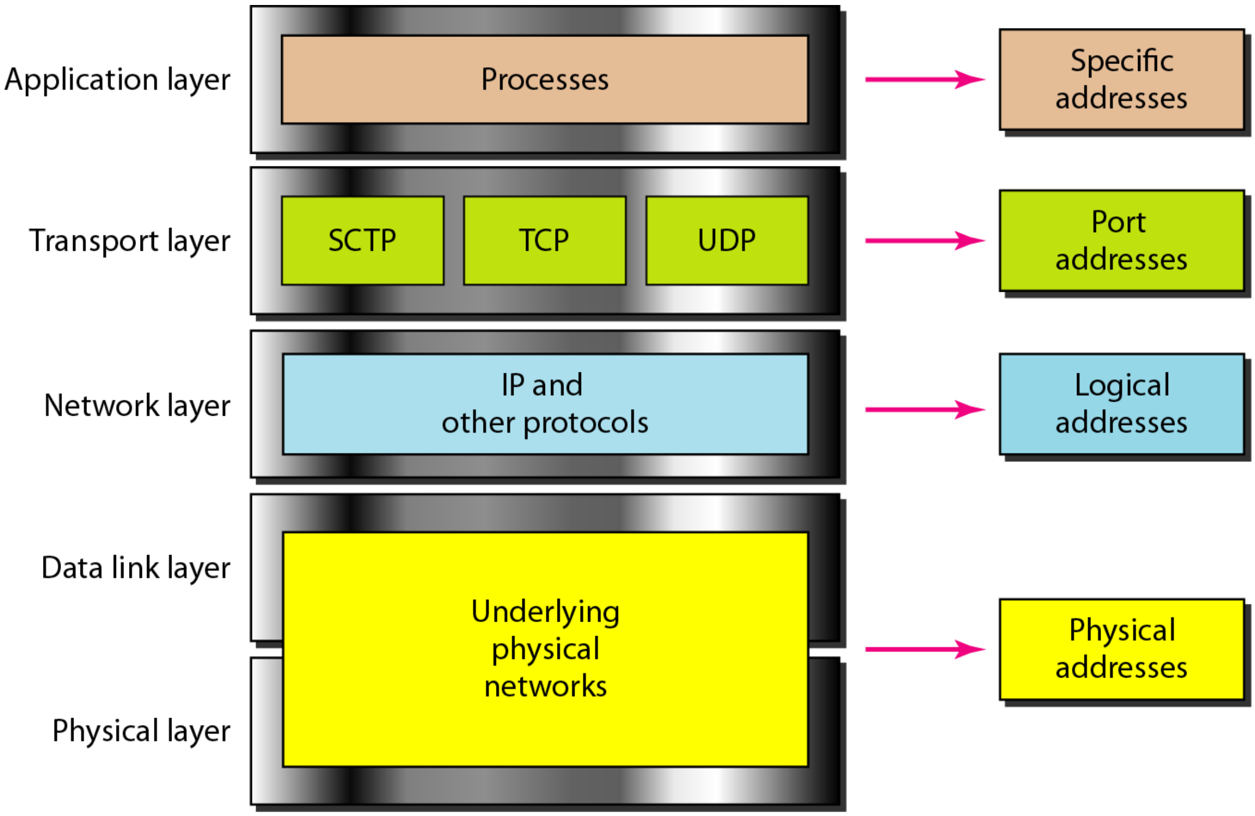


1.5 Addressing:



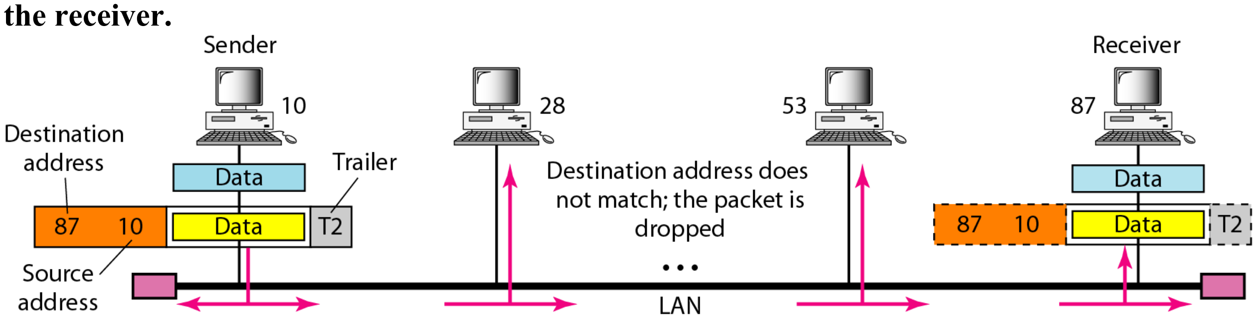
Four levels of addresses are used in an internet employing the TCP/IP Protocols:

i) Physical addresses
ii) Logical addresses
iii) Port addresses
iv) Specific address



Problems

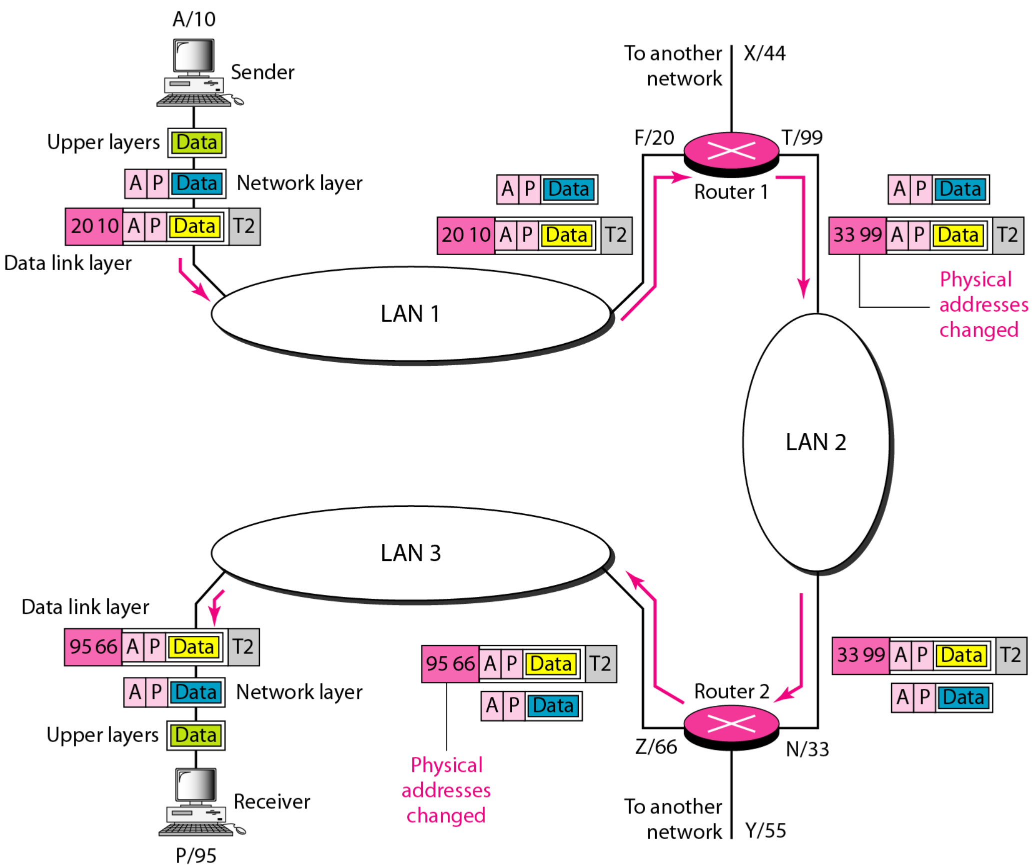
1.In Figure below a node with physical address 10 sends a frame to a node with physical address 87. The two nodes are connected by a link (bus topology LAN). As the figure shows, the computer with physical address 10 is the sender, and the computer with physical address 87 is the receiver.



2. ***Figure below shows a part of an internet with two routers connecting three LANs. Each device (computer or router) has a pair of addresses (logical and physical) for each connection. In this case, each computer is connected to only one link and therefore has only one pair of addresses. Each router, however, is connected to three networks (only two are shown in the figure). So each router has three pairs of addresses, one for each connection***

As we know local-area networks use a 48-bit (6-byte) physical address written as 12 hexadecimal digits; every byte (2 hexadecimal digits) is separated by a colon, as shown below:

07:01:02:01:2C:4B
A 6-byte (12 hexadecimal digits) physical address



Important questions:

1) List at-least 5 main responsibilities of Data link layer & Transport layer of OSI reference model.
2) What is layered architecture? Explain the interaction between the layers using a suitable diagram.
3) What are the responsibilities of the network layer in the internet model?
4) Name some services provided by the application layer in the internet model.
5) What is the difference between a port address, a logical address and a physical layer address?

Unit 2:

Data Link control

Hrs: 07

Syllabus:

Framing, Flow & Error control, Protocols, Noiseless channels & Noisy channels, HDLC.

Recommended readings:

Text Book: Data Communication & Networking, B Forouzan, 4e, TMH

Chapter 11 – page 307 to 346

2.1 Framing:

Framing in the data link layer separates a message from one source to a destination by adding a sender address & 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 & 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.

Frames can be of fixed or variable size.

i) Fixed-size framing:

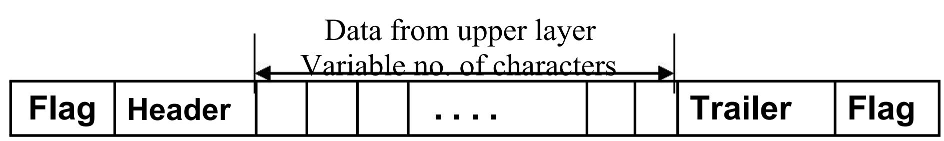
In this there is no need for defining the boundaries of the frames, the size itself can be used as a delimiter.
Ex: ATM wide area network which uses frames of fixed size called cells.

ii) Variable-size framing:

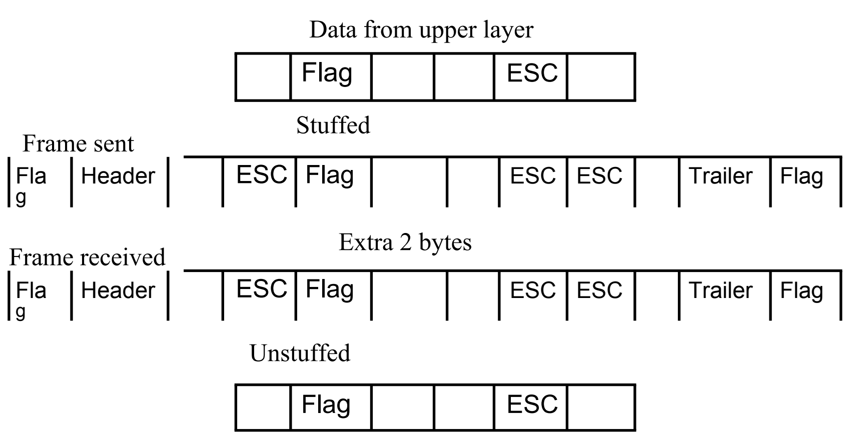
In this, we need a way to define the end of the frame and the beginning of the next. Ex: LAN

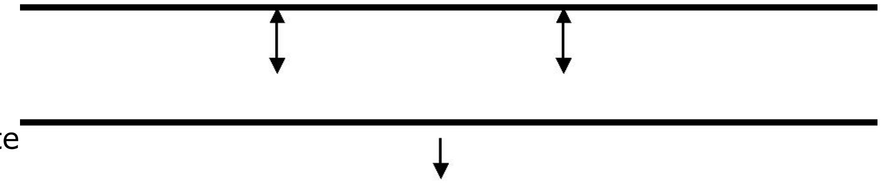
Historically 2 approaches were used for variable size framing: Character-oriented & bit-oriented.

a) Character-oriented approach:



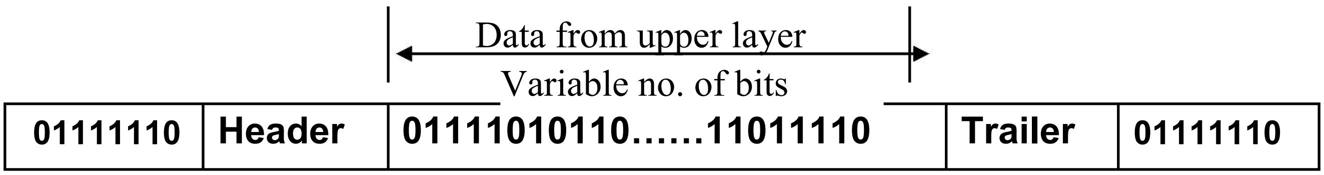
Character oriented protocol – Byte Stuffing & De-stuffing





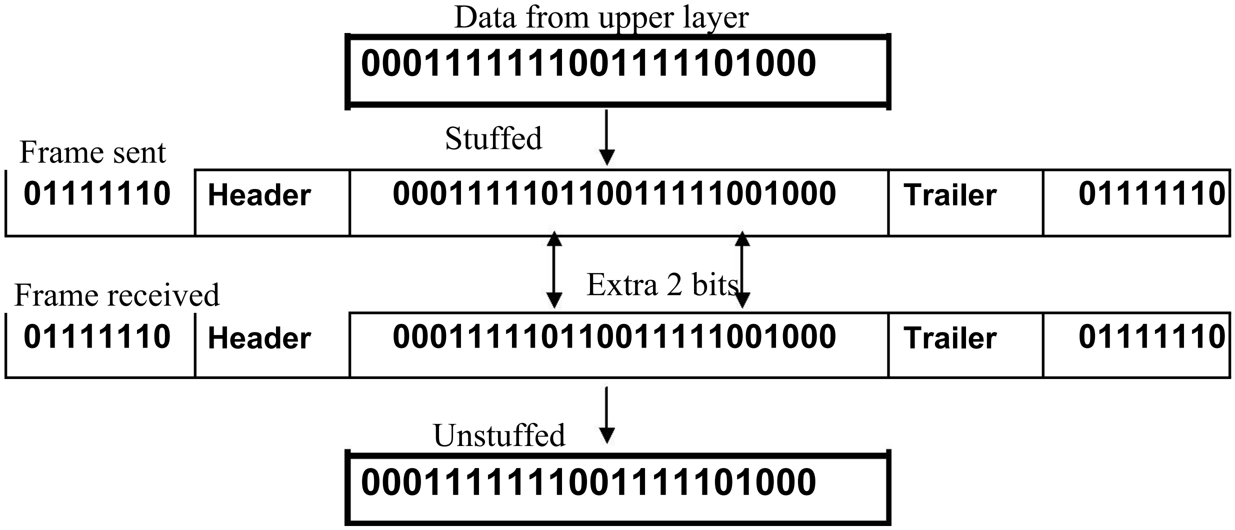
Data to upper layer

b) Bit-oriented approach:



Bit oriented protocol – Bit Stuffing & De-stuffing

Bit stuffing is the process of adding one extra 0 whenever five consecutive 1s follow a 0 in the data, so that the receiver doesn't mistake the pattern 01111110 for a flag



Data to upper layer

2.2 Flow & Error Control:

Flow Control:

- 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 flow of data must not be allowed to overwhelm the receiver.
- 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 acknowledgement.

Error Control:

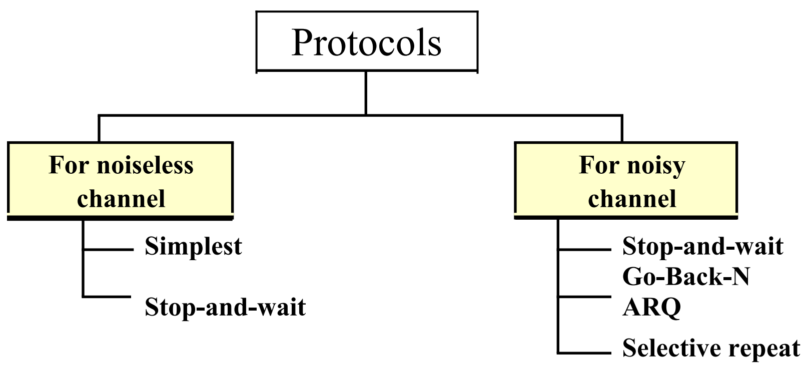
- Errors occur due to noises in the channel.

- Error control is both error detection & error correction.
- In the data link layer error control refers to methods of error detection and retransmission.
- To sender should add certain amount of redundant bits to the data, based on which the receiver will be able to detect errors.

2.3 PROTOCOLS:

- A protocol is a set of rules that govern data communication.
- A protocol defines what, how it is communicated, and when it is communicated.
- The key elements of a protocol are syntax, semantics & timing.
- Syntax refers to the structure or format of the data, i.e., the order in which they are presented.
- Semantics refers to the meaning of each section of bits.
- Timing refers to when data should be sent & how fast they can be sent.
- Protocols are implemented in software by using any of the common programming languages.

The protocols in the data link layer are classified as



2.4 Noiseless channels:

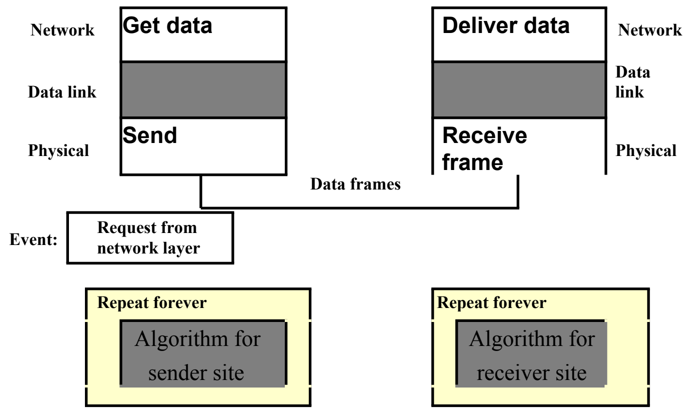
i) Simplest protocol:

Assumptions in this protocol are:

- Data transfer is unidirectional.
- Both sender & receiver network layers are always ready.
- Processing time can be ignored.
- Infinite buffer space is available.
- Frames are never damaged or lost.

Design:

Sender Receiver



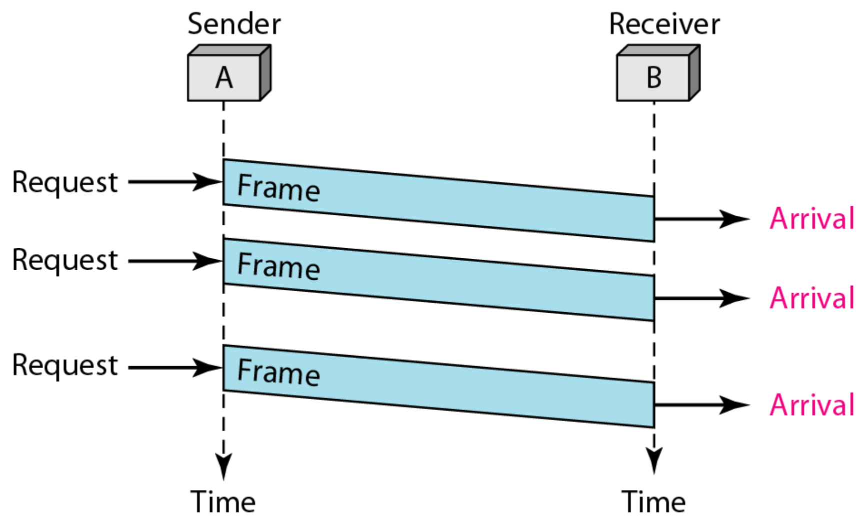
Sender-site algorithm for the Simplest Protocol:

```
While(true)                    //Repeat forever
{
  waitForEvent();              //Sleep until an event occurs
  if(event(requestTosend))     //there is a packet to send
    {
      GetData();               //get data from n/w layer
      MakeFrame();             //make a frame
      SendFrame();             //send the frame
    }
}
```

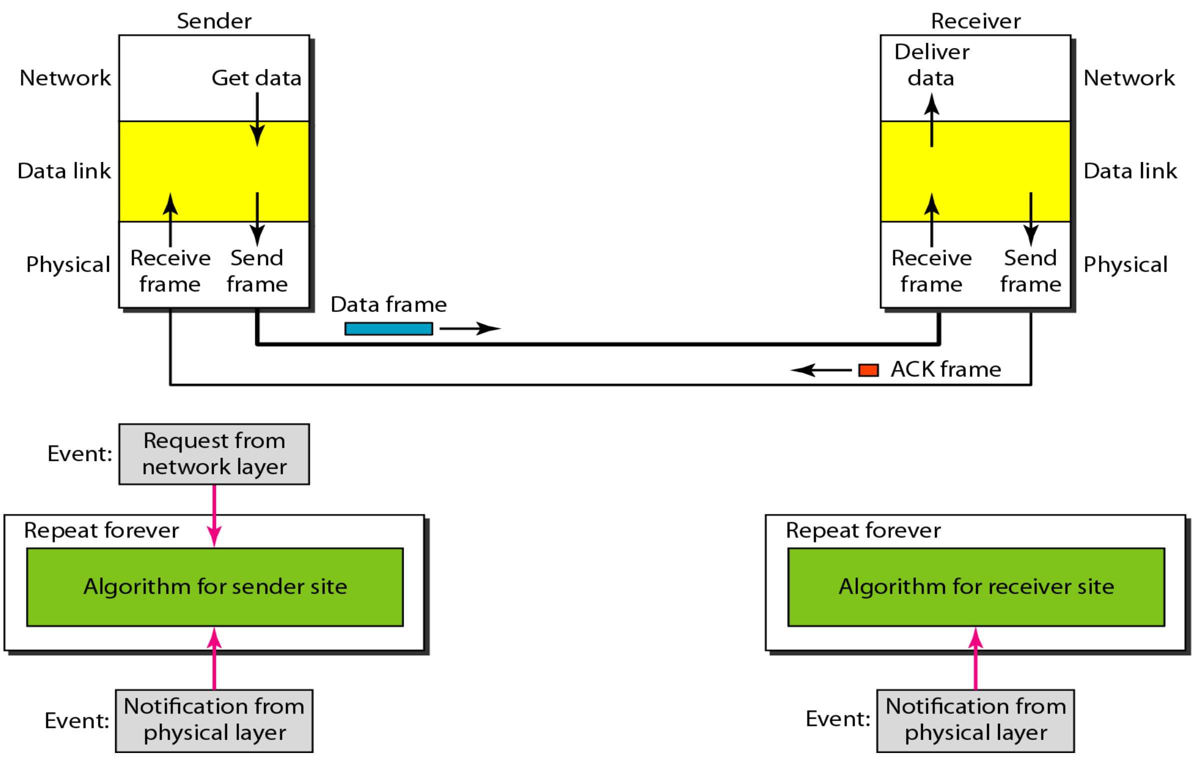
Receiver-site algorithm for the Simplest Protocol:

```
While(true)                    //Repeat forever
{
  waitForEvent();              //Sleep until an event occurs
  if(event(ArrivalNotification)) //data frame arrived
    {
      ReceiveFrame();          //receive frame from the physical layer
      ExtractData();           //extract data from a frame
      deliverData();           //deliver the data to the n/w layer
    }
}
```

Flow diagram to illustrate simplest protocol:



Design of Stop-and-Wait Protocol



ii) Stop - & - Wait Protocol:

Assumptions in this protocol are:

- Data transfer is unidirectional.
- Both sender & receiver network layers are always ready.

- Receiver doesn’t have enough storage space.
- Receiver is slower than sender in processing.
- Frames are never damaged or lost.

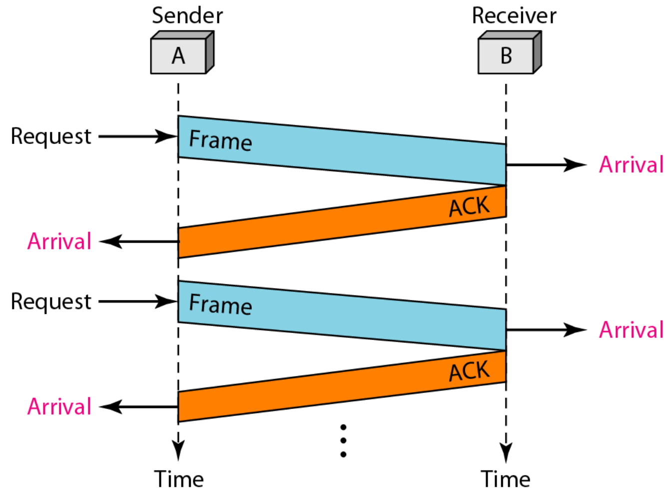
Sender-site algorithm for the Stop-&-wait Protocol:

```
While(true)                    //Repeat forever
Cansend = true                 //Allow the first frame to go
{
  waitForEvent();                      //Sleep until an event occurs
  if(event(requestTosend) AND cansend) //there is a packet to send
    {
      GetData();                       //get data from n/w layer
      MakeFrame();                     //make a frame
      SendFrame();                     //send the frame
      cansend = false;         //can’t send until ACK arrives
    }
  waitForEvent(); //Sleep until an event occurs if(Event(ArrivalNotification))
  //an ACK has arrived
   {
     ReceiveFrame();           //Receive the ACK frame
      cansend = true;
    }
}
```

Receiver-site algorithm for the Stop-&-Wait Protocol:

```
While(true)                    //Repeat forever
{
  waitForEvent();                      //Sleep until an event occurs
  if(event(ArrivalNotification)) //data frame arrived
    {
                               ReceiveFrame();                  //receive frame from the physical layer
      ExtractData();                   //extract data from a frame
                        deliverData();                          //deliver the data to the n/w layer
      SendFrame();                     //Send an ACK frame
     }
}
```

Flow diagram of Stop and Wait



2.5 Noisy channels:

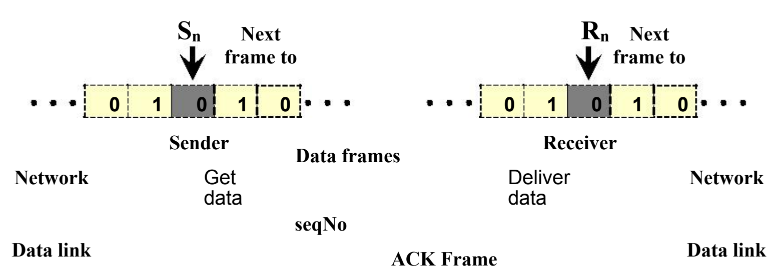
Assumptions are:

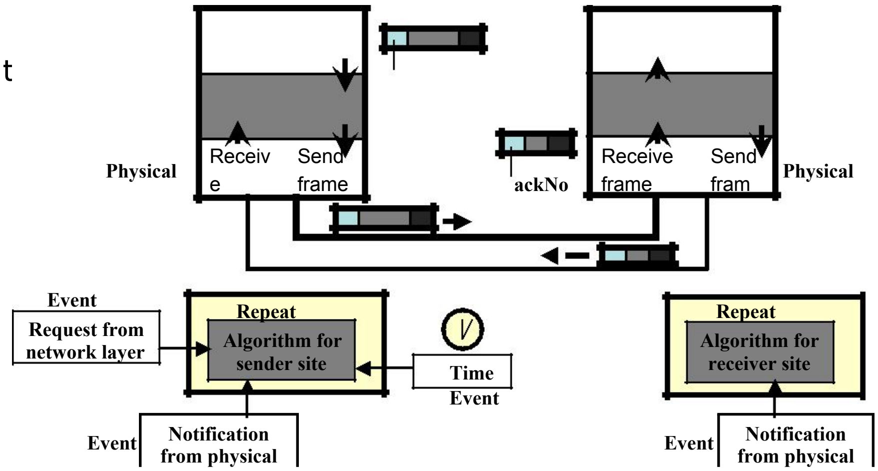
- Data transfer is unidirectional, but half-duplex link.
- Both sender & receiver network layers are always ready.
- Receiver doesn't have enough storage space.
- Receiver is slower than sender in processing.
- Frames are damaged or lost because of the noises in the channel.

ii) Stop - & - Wait Automatic Repeat Request (ARQ):

- Adds a simple error control mechanism to the Stop-&-Wait protocol.
- Error detection & retransmission is used for error control.
- Sending device keeps a copy of the last frame transmitted until acknowledgement for that frame is received.
- Data frames uses Sequence numbers, to avoid duplication of frames.
- Range of sequence numbers = 2m – 1.
- ACK frame uses Acknowledgement number, & always announces the sequence number of the next frame expected.
- Timers are used in case of loss of ACK frames.

Design:





Sender-site algorithm:

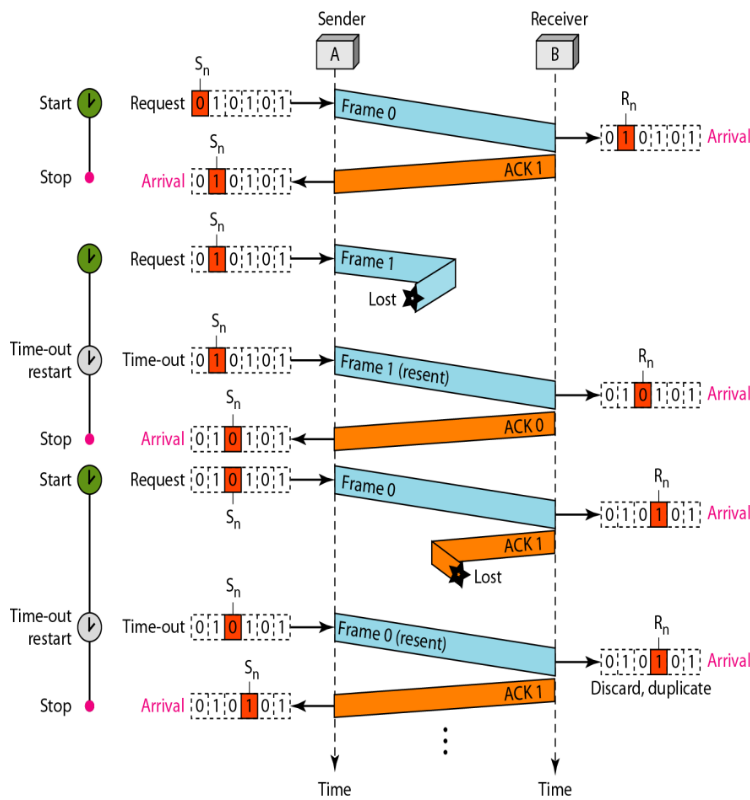
```
Sn = 0;                              //Frame 0 should be sent first
Cansend = true                       //Allow the first request to go
While(true)                          //Repeat forever
{
  waitForEvent();                              //Sleep until an event occurs
  if(event(requestTosend) AND cansend) //there is a packet to send
    {
      GetData();                     //get data from n/w layer
      MakeFrame(Sn);                 //make a frame
      StoreFrame(Sn);                        //copy of frame
      SendFrame(Sn);                 //send the frame
      StartTimer();
      Sn = Sn + 1;                             //Mod-2 addition
      cansend = false;               //can't send until ACK arrives
    }
  waitForEvent();                              //Sleep until an event occurs
if(Event(ArrivalNotification))                 //an ACK has arrived
    {
     ReceiveFrame(ackNo);                      //Receive the ACK frame
      if( not corrupted AND ackNo == Sn) // valid ACK
        {
          Stoptimer();
          purgeframe(Sn-1);
          cansend = true;
        }
     }
If(Event(Timeout))
  {
    StartTimer();
    ResendFrame(Sn-1);
   }
}
```

Receiver-site algorithm:

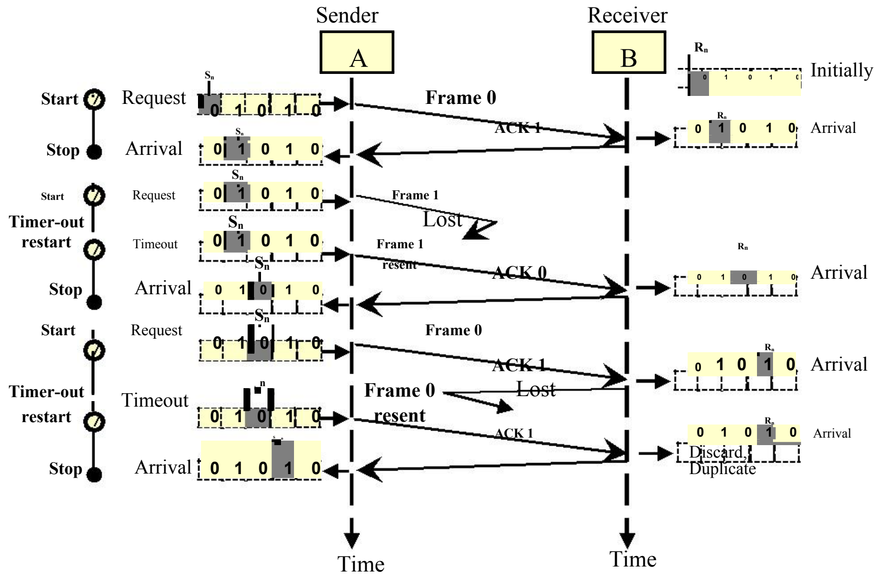
```
Rn = 0;                              //Frame 0 expected to arrive first
While(true)                          //Repeat forever
{
  waitForEvent();                              //Sleep until an event occurs
```

```
  if(event(ArrivalNotification)) //data frame arrived
    {
      ReceiveFrame();            //receive frame from the physical layer
      if(Corrupted(Frame))
         sleep();
      if(seqNo == Rn)
        {
          ExtractData();                  //extract data from a frame
          DeliverData();                  //deliver the data to the n/w layer
          Rn = Rn +1;
        }
       SendFrame();                       //Send an ACK frame
    }
}
```

Flow Diagram



Flow diagram:

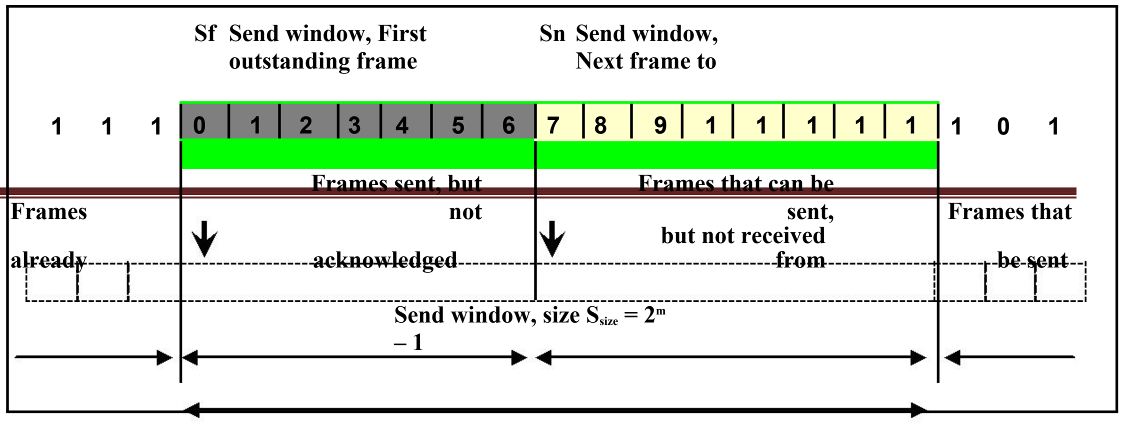


ii) Go-Back-N ARQ Protocol:

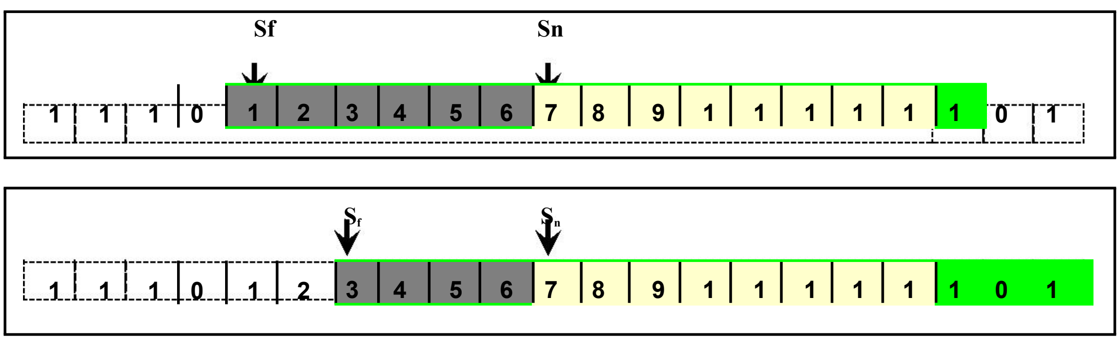
- Bandwidth-Delay product.
- Pipelining.
- Sequence numbers range from 0 to 2m – 1.
- Sliding window is an abstract concept that defines the range of sequence numbers that is the concern of the Sender & Receiver.
- Timers – timer for the first outstanding frame always expires first, we resend all outstanding frames when this timer expires.
- Acknowledgement – sends ACK for safe and in order frames, receiver will be silent for corrupted & out of order frames.
- Resending a frame – when the timer expires, the sender resends all the outstanding frames.

Send window for Go-Back-N ARQ

a) Send window before sliding

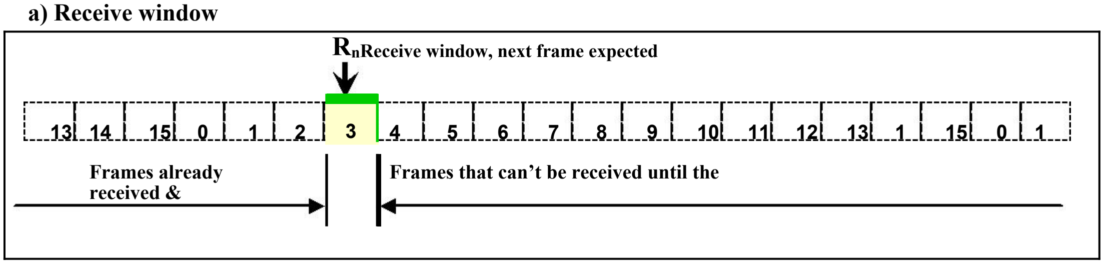


b) Send window after sliding



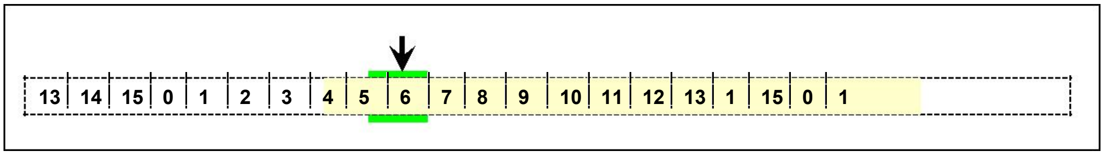
Receive window for Go-Back-N ARQ

a) Receive window



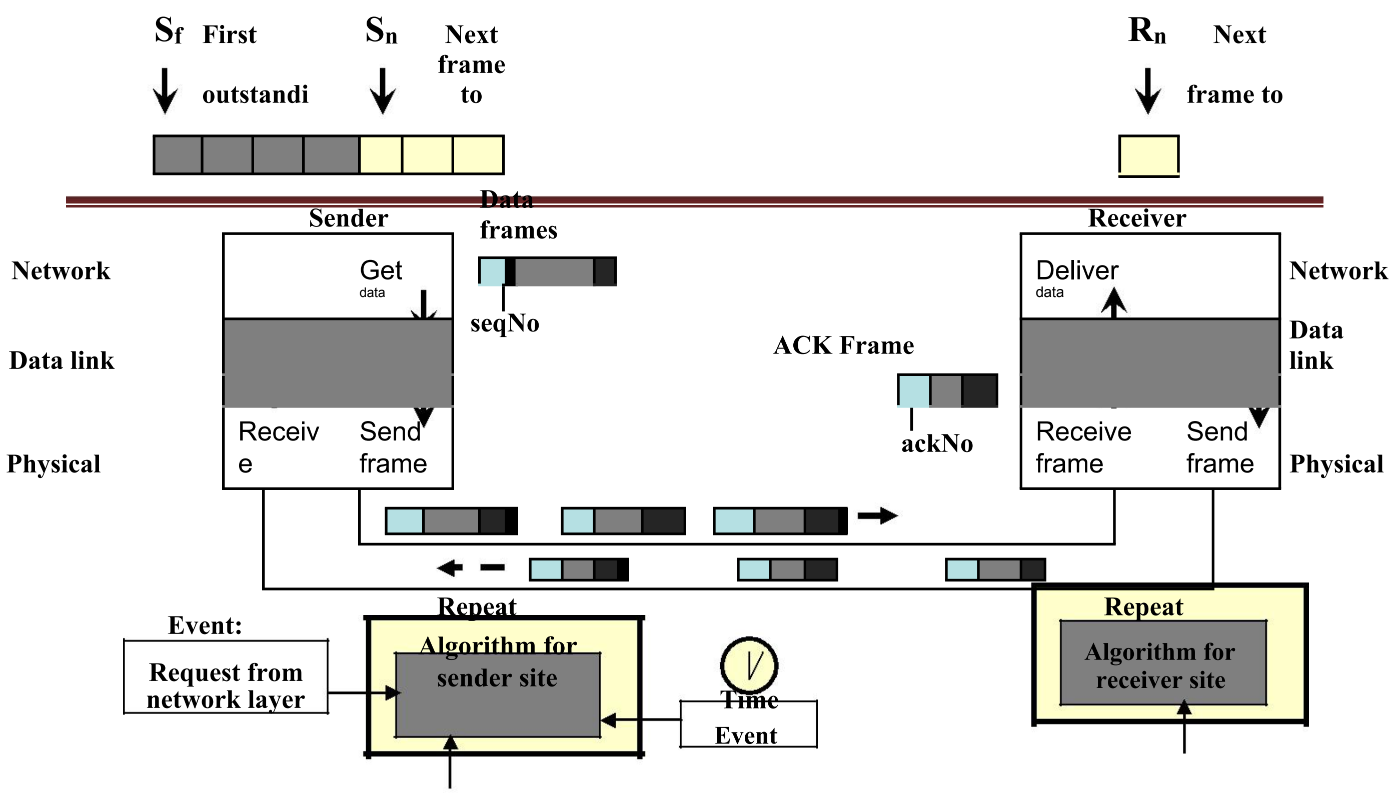
b) Window after sliding

R_n



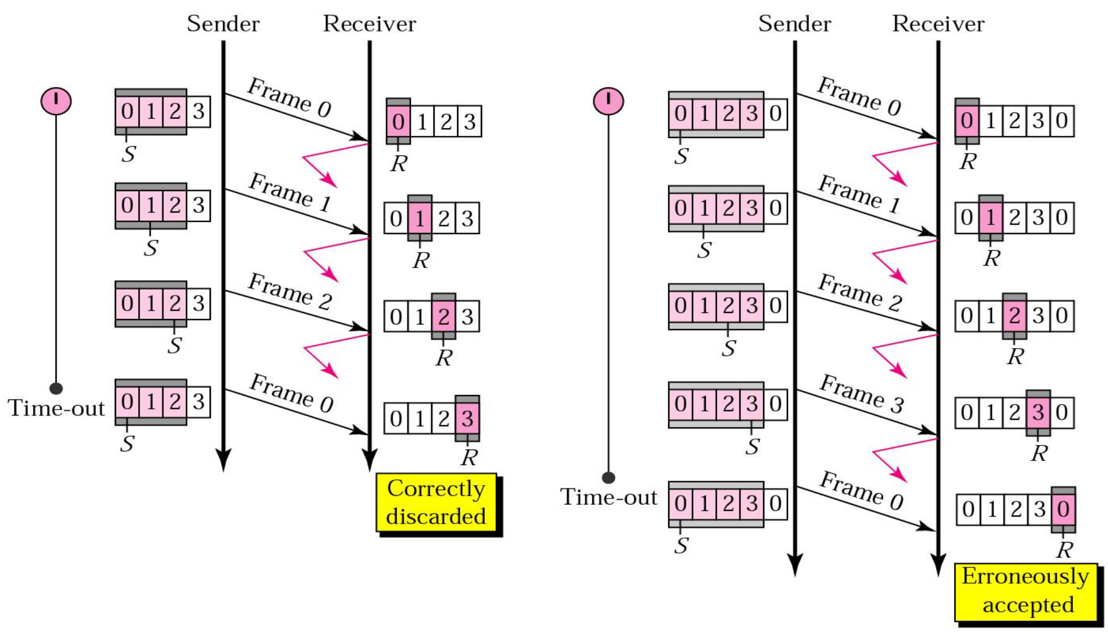
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 1 15 0 1

Design:



Sf First outstandi
Sn Next frame to
Rn Next frame to
Sender
Data frames
Receiver
Network
Get data
seqNo
Deliver data
Network
Data link
ACK Frame
Data link
Physical
Receiv e
Send frame
ackNo
Receive frame
Send frame
Physical
Event:
Request from network layer
Repeat
Algorithm for sender site
Time Event
Repeat
Algorithm for receiver site

Window size for Go-Back-N ARQ:



a. Window size < 2^m b. Window size = 2^m

Sender-site algorithm:

```
Sw = 2^m – 1;
Sf = 0;
Sn = 0;                        //Frame 0 should be sent first
While(true)                    //Repeat forever
{
  waitForEvent();                         //Sleep until an event occurs
  if(event(requestTosend)) //there is a packet to send
    {
      if(Sn – Sf>= Sw)         //If window is full
         Sleep();
      GetData();                          //get data from n/w layer
      MakeFrame(Sn);           //make a frame
      StoreFrame(Sn);                     //copy of frame
      SendFrame(Sn);           //send the frame
      StartTimer();
      Sn = Sn + 1;                        //Mod-2 addition
      if(timer not running);              //can't send until ACK arrives
         StartTimer();
```

```
      }
  if(Event(ArrivalNotification))                //an ACK has arrived
     {
       Receive (ACK);               //Receive the ACK frame
        if(corrupted (ACK))
          Sleep();
        if((ackNo>Sf&& (ackNo<= Sn)))
        While(Sf<= ackNo)
           {
            Purgeframe(Sn-1);
            Sf = Sf + 1;
            }
         StopTimer();
      }
  If(Event(Timeout))
    {
      StartTimer();
      Temp = Sf;
      While(Temp <Sn)
         {
           SendFrame(Sf);          //sendframe(temp);
            Sf = Sf + 1;            //temp = temp+1;
          }
     }
  }
```

Receiver-site algorithm :

```
Rn = 0;                              //Frame 0 expected to arrive first
While(true)                    //Repeat forever
{

  waitFor    //Sleep until
  Event      an event
  ();        occurs
  if(event(
  Arrival
  Notific    //data frame
  ation))    arrived
    {
      Receive
      F
      ra
      m   //receive frame
      e(   from the physical
      );   layer
      if(Corru
      pt
```

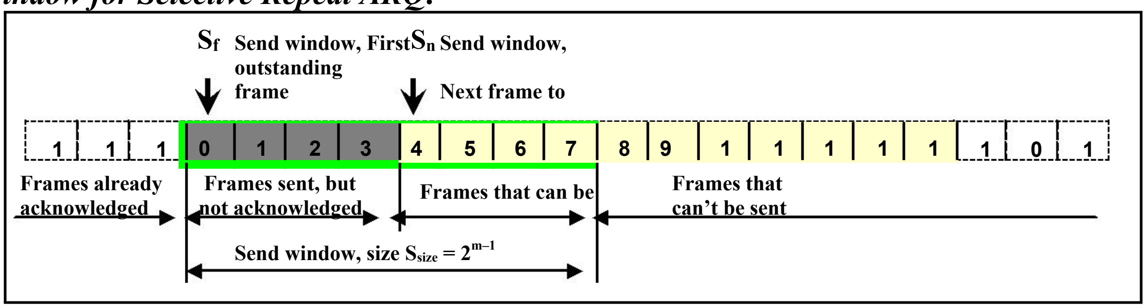
```
  e
  d(
  F
  ra
  m
  e)
  )
    sleep();
  if(seqN
  o
  =
  =
  R
  n)
    {
     Deliver
     D
     a
     t
     a
     (    //deliver the
     )    data to the n/w
     ;    layer
     Rn
     =
     R
     n
     +
     1
     ;//Slide window
     SendA
     C
     K
     (
     R
     n
     )
     ;
    }
 }
}
```

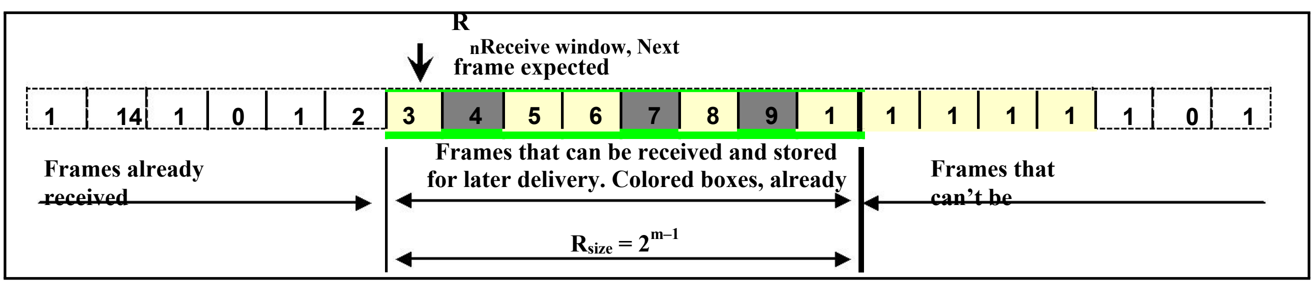
Stop-and-Wait ARQ is actually a Go-Back-N ARQ in which there are only two sequence numbers and the send window size is 1. i.e.,
$m = 1; 2^m - 1 = 1$

iii) Selective Repeat ARQ Protocol:

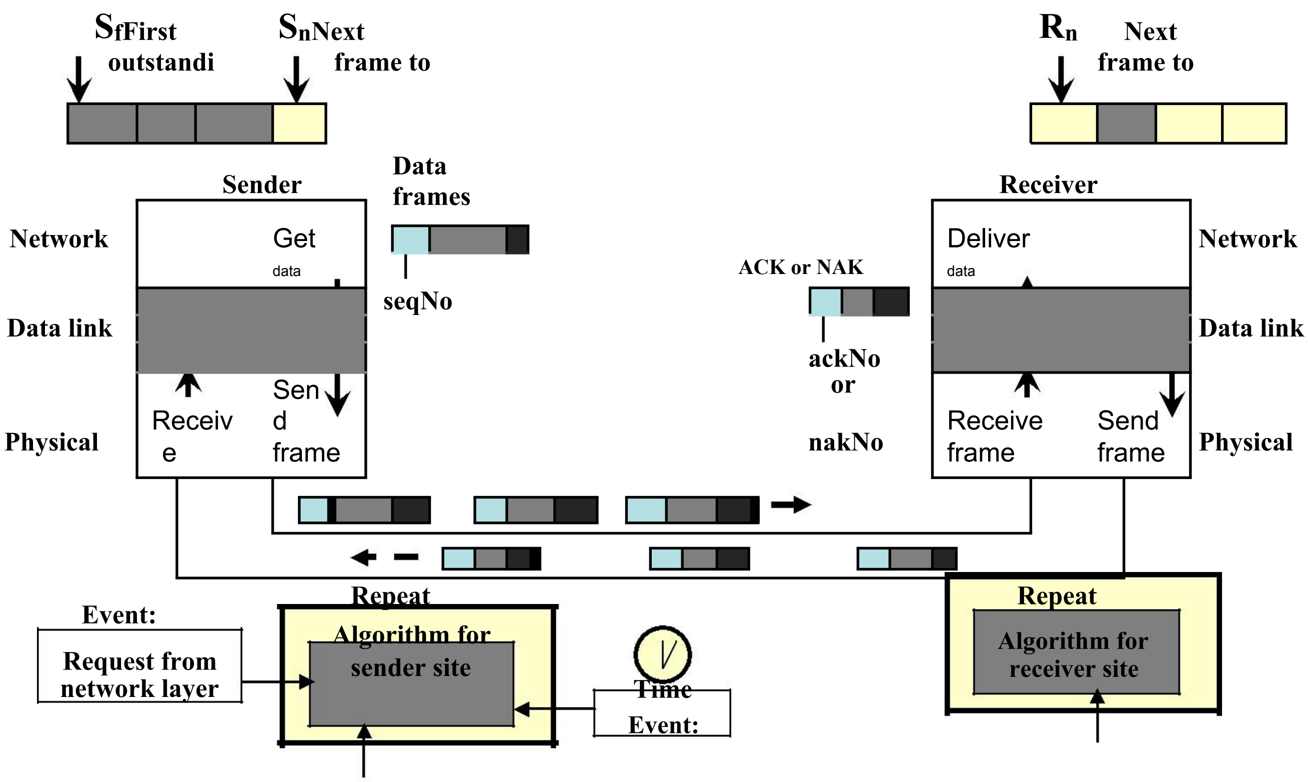
- In a noisy link frame has a higher probability of damage, means resending of multiple frames.
- This uses more bandwidth & slows down the transmission.
- Instead of sending N frames when just one frame is damaged, a mechanism is required to resend only the damaged frame.
- Sequence numbers range from 0 to 2m – 1.
- Both send & receive window size = 2(m – 1).

Send window for Selective Repeat ARQ:



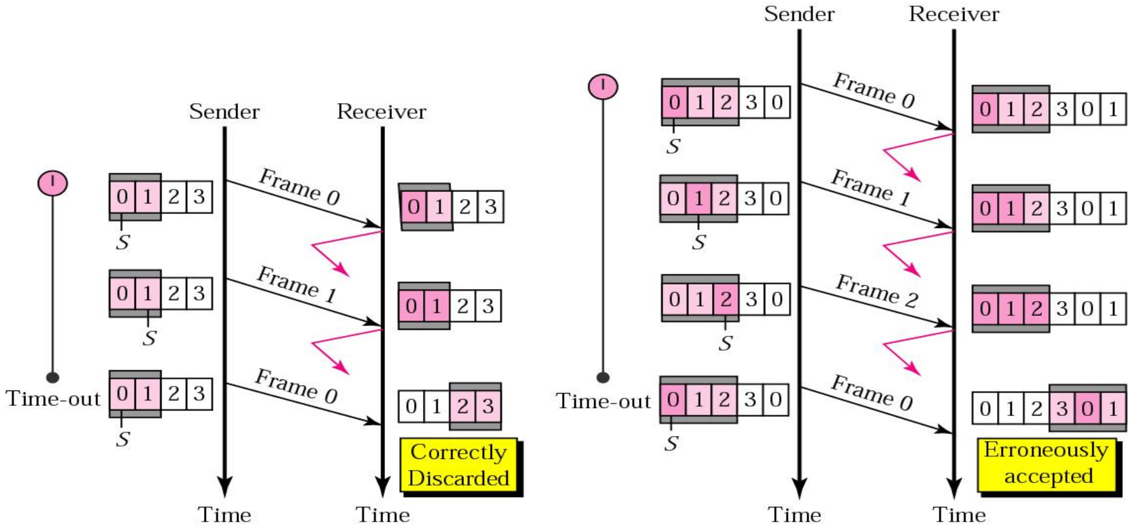


Design:



SfFirst
outstandi
SnNext
frame to
Rn
Next
frame to
Sender
Data
frames
Receiver
Network
Get
data
seqNo
ACK or NAK
Deliver
data
Network
Data link
ackNo
or
nakNo
Data link
Physical
Receiv
e
Sen
d
frame
Receive
frame
Send
frame
Physical
Event:
Request from
network layer
Repeat
Algorithm for
sender site
Time
Event:
Repeat
Algorithm for
receiver site

Window size for Selective Repeat ARQ:



a. Window size = 2^{m-1} b. Window size > 2^{m-1}

Sender-site algorithm:

```
Sw = 2m-1 ;
Sf = 0;
Sn = 0;                        //Frame 0 should be sent first
While(true)                    //Repeat forever
{
  waitForEvent();                      //Sleep until an event occurs
  if(event(requestTosend)) //there is a packet to send
    {
      if(Sn – Sf>= Sw)                 //If window is full
         Sleep();
      GetData();                       //get data from n/w layer
      MakeFrame(Sn);                   //make a frame
      StoreFrame(Sn);                       //copy of frame
      SendFrame(Sn);           //send the frame
      Sn = Sn + 1;
      StartTimer(Sn);
}
if(Event(ArrivalNotification))         //an ACK has arrived
    {  Receive (frame);                //Receive the ACK frame
       if (corrupted (frame))
         Sleep();
        if (FrameType == NAK)
          if (nakNo between Sf&Sn)
```

```
            { resend(nakNo);
              StartTimer(nakNo);
            }
          if (FrameType == ACK)
            if (ackNo between Sf&Sn)

              {   While (Sf<= ackNo)
                      {
                       Purge (Sf);
                       StopTimer (Sf);
                       Sf = Sf + 1;
                      }
              }
      }
If (Event (Timeout (t)))
  {
    StartTimer(t);
    SendFrame(t);
  }
}
```

Receiver-site algorithm :

```
Rn = 0;                  //Frame 0 expected to arrive first
Naksent = false;
AckNeeded = false;
Repeat( for all slots)
    Marked (slot) = false;

While(true)                         //Repeat forever
{
  waitForEvent();                        //Sleep until an event occurs
  if(event(ArrivalNotification))         //data frame arrived
    {
      ReceiveFrame();
      if(Corrupted(Frame) && NOT NakSent)
        {
         SendNAK(Rn);
         Naksent = true;
         sleep();
        }
if(seqNo<>Rn&& NOT Naksent)
   {  SendNAK(Rn);
      Naksent = true;
      if((seqno        in        window)        &&
```

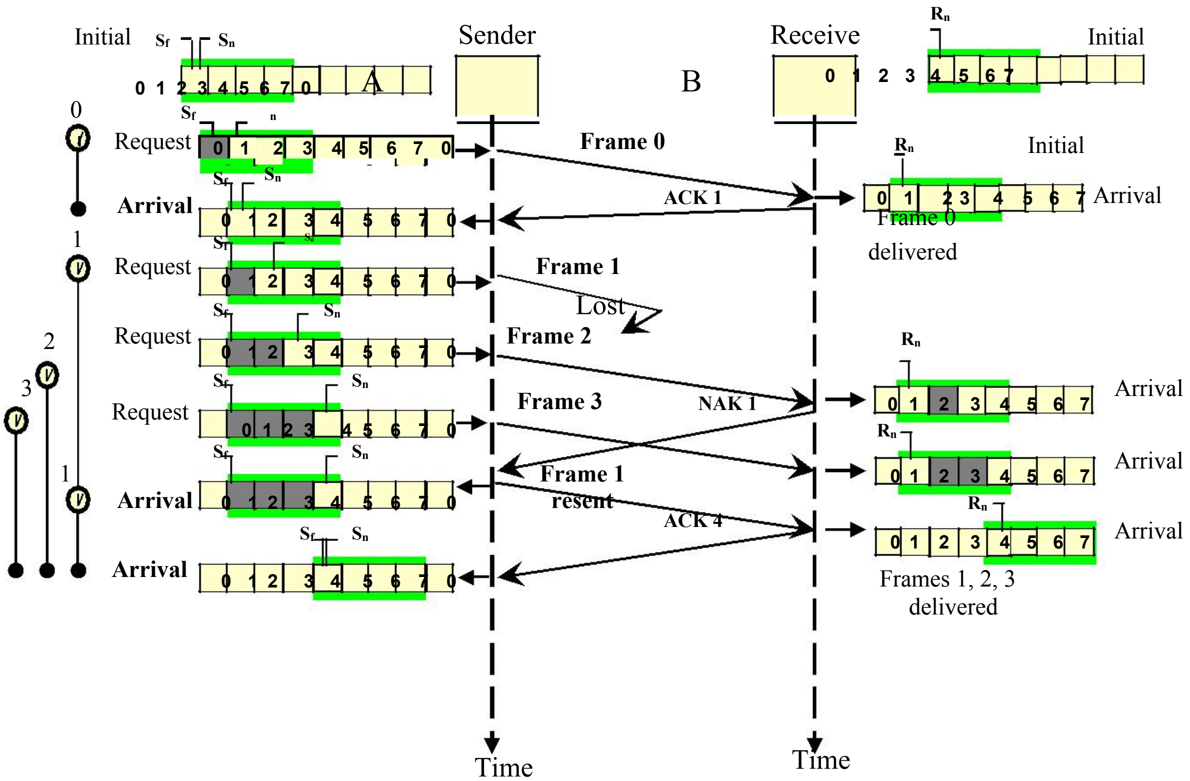
```
(!Marked(seqno))) { StoreFrame(seqNo);
    Marked(seqNo) = true;
    while(Marked(Rn))
       { DeliverData(Rn);                  //deliver the data to the n/w layer
          purge(Rn);
          Rn = Rn +1;             //Slide window
          AckNeeded = true; }
     if(AckNeeded)
```

```
                { SendACK(Rn);
                  AckNeeded = false;
                  NakSent = false;       }
            }
    }
    }
}
```

Piggybacking

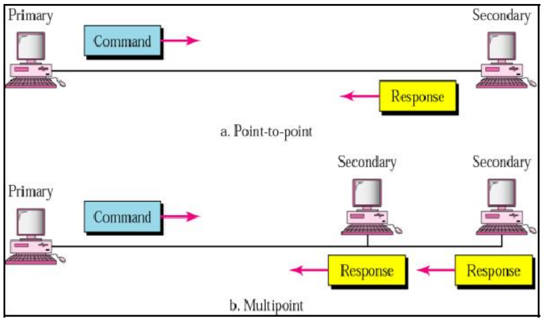
- When the data transfer is bidirectional, i.e., from node A to node B and from node B to node A, the control information also needs to flow in both the directions.
- Efficiency can be improved if the control information can be passed to the other end along with the data itself which is flowing to that end. This concept is called as piggybacking.

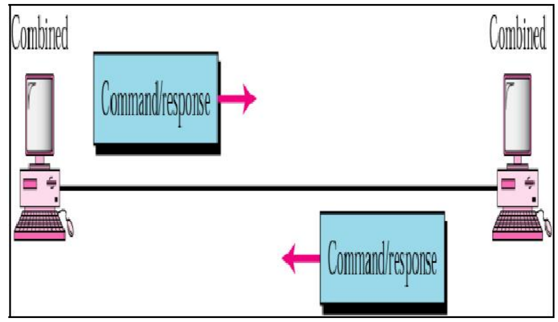
Flow diagram:



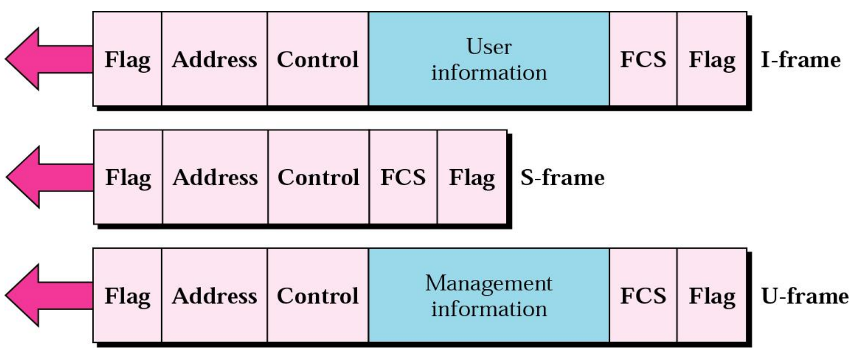
2.6 High-level Data Link Control (HDLC):

- HDLC is a bit-oriented protocol for communication over point-to-point and multipoint links.
- It implements the ARQ mechanisms.
- HDLC provides two common transfer modes.

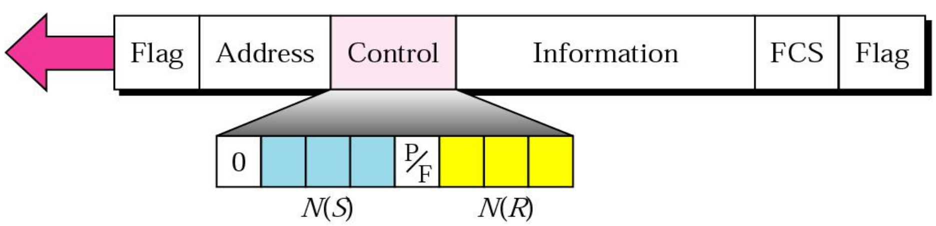




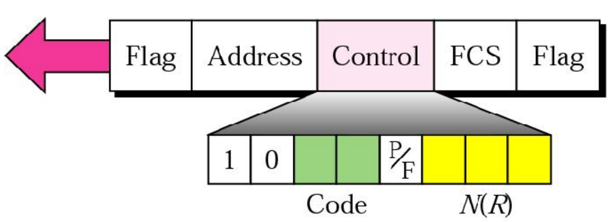
Frame format :



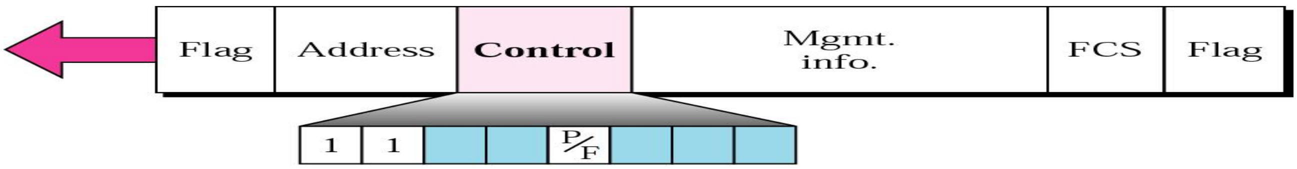
Information frames (I-Frames) :



Supervisory frames (I-Frames) :



Unnumbered frames (I-Frames) :



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

Important questions:

1. What are the three protocols considered for noisy channels?
2. Define framing and the reason for its need.
3. Compare and contrast the G0-Back-N ARQ protocol with Selective repeat ARQ.
4. Briefly describe the services provided by the data link layer.
5. Define piggybacking and its uses.
6. Compare and contrast byte-stuffing and bit-stuffing. Which technique is used in byte-oriented protocols? Which technique is used in bit-oriented protocols?
7. Which all the protocols of data link layer uses pipelining concept?
8. Compare and contrast flow control and error control.
9. Compare and contrast HDLC with PPP. Which one is byte-oriented; which one is bit-oriented?
10. Explain the reason for moving from stop-and-wait ARQ protocol to the Go-Back-N ARQ protocol.