Data Link Layer

What is DLL(Data Link Layer)

The Data Link Layer is the second layer in the OSI model, above the Physical Layer, which ensures that the error free data is transferred between the adjacent nodes in the network. It breaks the datagram's passed down by above layers and converts them into frames ready for transfer. This is called Framing. It provides two main functionalities

- Reliable data transfer service between two peer network layers
- Flow Control mechanism which regulates the flow of frames such that data congestion is not there at slow receivers due to fast senders.

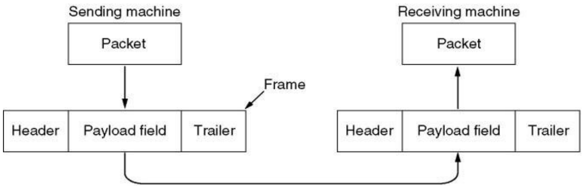
DATA LINK LAYER DESIGN ISSUES:-

For effective data communication between two directly connected stations, the data link layer has to carry out a number of s ific functions.

a) Services Provided to the Network Layer
b) Framing
c) Flow Control
d) Error Control
e) Addressing
f) Link Management

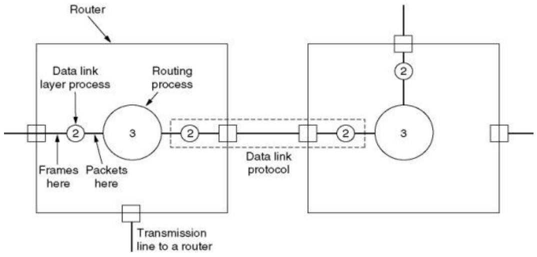
a) Services Provided to the Network Layer:
The primary service is transferring data from Network layer on source machine to Network layer on destination machine.
b) Framing: The source machine sends data in block called frames to the destination machine. The starting and ending on each frame must be recognized by the destination machine
c) Flow control: The source machine should not send data frames at a rate faster than the destination machine can accept them
d) Error Control: the errors made in bits during transmission from source to destination machines must be detected and corrected
e) Addressing: On network(LAN), the identity of individual machines must be s ified while transmitting the data frames.
f) Link Management: The initiation, maintenance and termination of the link between the source and destination is required for effective exchange of data.

Relationship between packets and frames



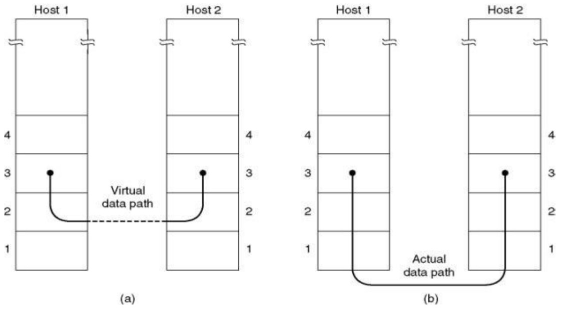
Placement of Data Link Protocol:

1) Physical Layer Entry
2) DLL process
3) Routing process



Services Provided to the Network Layer:

The primary service is transferring data from Network layer on source machine to Network layer on destination machine. The virtual path in fig (a) is not the actual path, the fig(b) shows the actual path .



(a) Virtual path (b) Actual path

Types of services provided :

- Unacknowledged connection less service (WAN)
- Acknowledged connection less service (Wireless)
- Acknowledged connection oriented service (LAN)

Acknowledged Connection-oriented service:

- Each frame received exactly once
- All frames received in the right order
- each frame numbered, DLL guaranties reception of all frames sent
- reliable bit stream for NW layer
- 3 distinct phases:
    - Establishment of connection
    - Data transfer
    - Release of connection

Acknowledged connection less service (Wireless):

- No connection used but each frame individually added.
- sender knows whether frame received safely or not.
- useful over unreliable links – wireless links!
- Acknowledgement for each packet
- Resending
- Ack is optimisation; also transport layer can handle errors

Unacknowledged connection less service (WAN):

- no connection established beforehand or released afterwards.
- Independent frames
- Error rate should be low
- If a frame is lost due to noise then no Recovery
- Used on LANs

FRAMING:

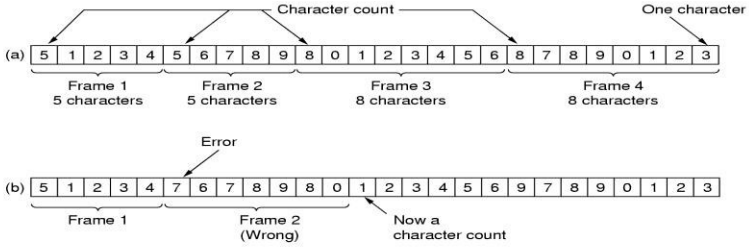
Since the physical layer merely accepts and transmits a stream of bits without any regard to meaning or structure, it is upto the data link layer to create and recognize frame boundaries. This can be accomplished by attaching s ial bit patterns to the beginning and end of the frame. The four framing methods that are widely used are

- Character count
- Starting and ending characters, with character stuffing
- Starting and ending flags, with bit stuffing
- Physical layer coding violations

Character Count

This method uses a field in the header to s ify the number of characters in the frame. When the data link layer at the destination sees the character count, it knows how many characters follow, and hence where the end of the frame is.

The disadvantage is that if the count is garbled by a transmission error, the destination will lose synchronization and will be unable to locate the start of the next frame. So, this method is rarely used.



Out of synchronisation if error

Character stuffing

In the second method, each frame starts with the ASCII character sequence DLE STX and ends with the sequence DLE ETX.(where DLE is Data Link Escape, STX is Start of TeXt and ETX is End of TeXt.) This method overcomes the drawbacks of the character count method. If the destination ever loses synchronization, it only has to look for DLE STX and DLE ETX characters.

If however, binary data is being transmitted then there exists a possibility of the characters DLE STX and DLE ETX occurring in the data. Since this can interfere with the framing, a technique called character stuffing is used. The sender's data link layer inserts an ASCII DLE character just before the DLE character in the data. The receiver's data link layer removes this DLE before this data is given to the network layer.

DLE	STX	A	B	C	D	DLE	ETX

A	DLE	C

DLE	STX	A	DLE	DLE	C	DLE	ETX

- Example: If the frame contained ``A B DLE D E DLE", the characters transmitted over the channel would be ``DLE STX A B DLE DLE D E DLE DLE DLE ETX".

Disadvantage: character is the smallest unit that can be operated on; not all architectures are byte oriented.

Bit stuffing

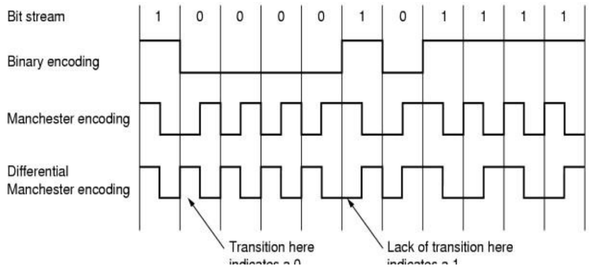
The third method allows data frames to contain an arbitrary number of bits and allows character codes with an arbitrary number of bits per character. At the start and end of each frame is a flag byte consisting of the s ial bit pattern 01111110 . Whenever the sender's data link layer encounters **five consecutive 1s** in the data, it automatically stuffs a zero bit into the outgoing bit stream. This technique is called bit stuffing.

When the receiver sees five consecutive 1s in the incoming data stream, followed by a zero bit, it automatically destuffs the 0 bit. The boundary between two frames can be determined by locating the flag pattern.

S ial bit pattern for start / end of frame 01111110
Whenever data contains 5 consecutive ones insert
0 Example:
– 0110111111111111110 NWL A
– 01101111101111101110 Physical
– 0110111111111111110 NWL B

Physical layer coding violations

The final framing method is physical layer coding violations and is applicable to networks in which the encoding on the physical medium contains some redundancy. In such cases normally, a 1 bit is a high-low pair and a 0 bit is a low-high pair. The combinations of low-low and high-high which are not used for data may be used for marking frame boundaries.


Bit stream
1 0 0 0 0 1 0 1 1 1 1
Binary encoding
Manchester encoding
Differential Manchester encoding
Transition here indicates a 0
Lack of transition here indicates a 1

Use no transition in a slot (= coding violation) as start of frame

Error control :

- How does a sender know that all packets are correctly received?
  - Sender requests ACK packet by receiver
  - A +ve ACK for delivered safely
  - A –ve ACK for frame retransmission.
- Packet lost or not recognized at receiver or ACK lost?
  - Timer at sender: to resend a packet, Timer starts with packet.
- How to handle duplicates or out of order received packets? i.e transmitting a frame multiple time.
  - So Sequence number in packet and ACK packet are added
- Optimisation: NACK packet
  - Inform sender that something strange happened

Flow Control:

- Even it is error-free channel, receiver will simply unable to handle the frames as they arrive and will start loss some.
- Flow control refers to set of methods to restrict the amount of data, that sender can send before waiting for ACK
- How to prevent a sender to overload the receiver with packets

is Two approaches are commonly used

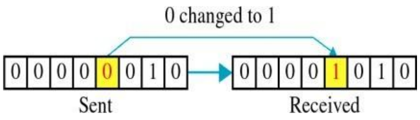
1. Feedback based Flow control:
   The receiver gives he permission to send more data
2. Rate based Flow control:
   It has a built in mechanism that limits the rate at which senders may transmit data, without feedback from receiver.

ERROR DETECTION AND CORRECTION:-

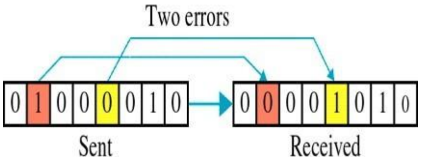
Data can be corrupted during transmission. For reliable communication, errors must be detected and corrected. Transmission errors are caused by electrical interference from natural sources, such as lightning, as well as from man-made sources, such as motors, generators, power lines, and fluorescent lights.

Types of errors: a) Single Bit error b)Burst error

Single-bit error : In a single –bit-error, a ‘0’ is changed to ‘1’ or a ‘1’ to ‘0’



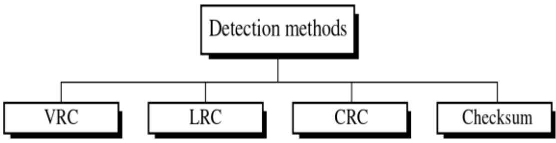
Burst error : The term burst error means that two or more bits in the data unit have changed from 1 to 0 or from 0 to 1.Burst errors does not necessarily mean that the errors occur in consecutive bits, the length of the burst is measured from the first corrupted bit to the last corrupted bit



ERROR DETECTION:

Before correct the error’s they has to detect first. Error detection means to decide whether the received data is correct or not without having a copy of the original message. Error detection **uses the concept of redundancy**, **which means** adding extra bits for detecting errors at the destination.

One method is : Each data unit is sent multiple times, usually twice. At the receiver , the two units are compared, and if they are the same, it is assumed that no transmission errors have occurred. It may be sometimes accurate but, Transmission time is double, it takes time for every bit comparison. So including extra information(redundant) is good than this.



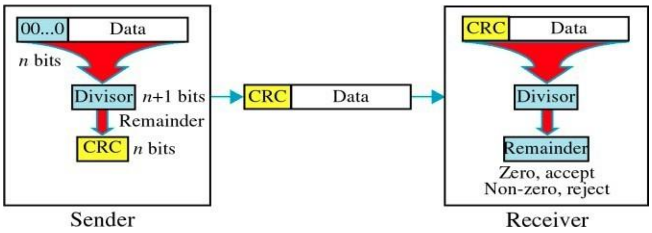
Four types of redundancy checks are used

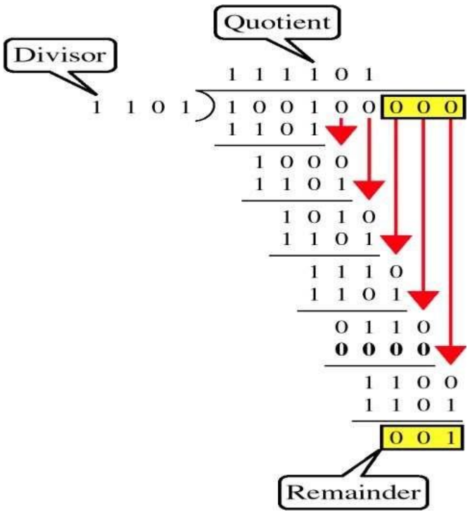
Cyclic Redundancy Check (CRC):

- CRC is a Powerful error detection scheme. It is based on binary division.

Algorithm:

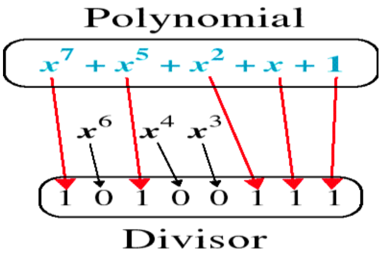
1. A string of n 0's bits is appended to data unit. The n is '1' less than the number of bits in divisor
2. The Data + no. of zero's is divided by the divisor using binary division, then the remainder is CRC
3. The Data + obtained CRC is send to receiver
4. The Receiver receives the Data + CRC and divided by same divisor
5. If remainder is '0' then the Data is accepted.





polynomial

$$x^7 + x^5 + x^2 + x + 1$$



Standard Polynomials

CRC-12

$$x^{12} + x^{11} + x^3 + x + 1$$

CRC-16

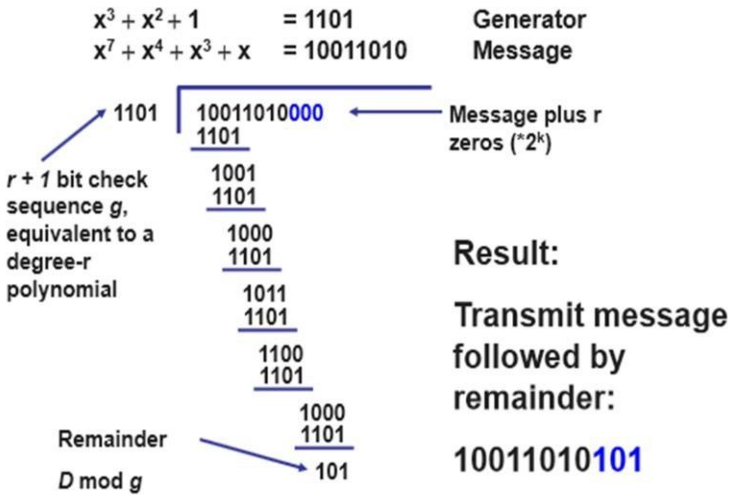
$$x^{16} + x^{15} + x^2 + 1$$

CRC-ITU

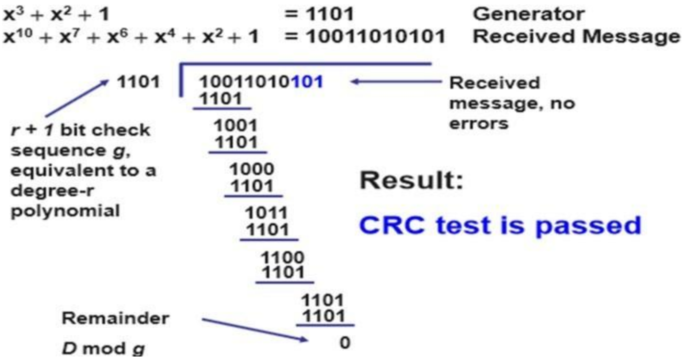
$$x^{16} + x^{12} + x^5 + 1$$

CRC-32

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$


$x^3 + x^2 + 1$ = 1101 Generator
$x^7 + x^4 + x^3 + x$ = 10011010 Message
1101
10011010000
Message plus r zeros (*2^k)
r + 1 bit check sequence g, equivalent to a degree-r polynomial
1101
1001
1101
1000
1101
1011
1101
1100
1101
1000
1101
101
Remainder
D mod g
Result:
Transmit message followed by remainder:
10011010101

AT SENDER


$x^3 + x^2 + 1$ = 1101 Generator
$x^{10} + x^7 + x^6 + x^4 + x^2 + 1$ = 10011010101 Received Message
1101
10011010101
Received message, no errors
r + 1 bit check sequence g, equivalent to a degree-r polynomial
1101
1001
1101
1000
1101
1011
1101
1100
1101
1101
1101
0
Remainder
D mod g
Result:
CRC test is passed

AT RECEIVER

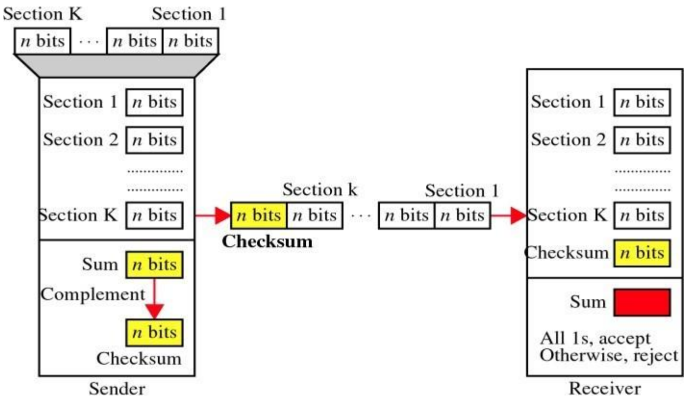
Checksum:

Checksum is based on the concept of redundancy. At sender side

- The checksum generator subdivides the data unit into equal segments of ‘n’ bits.
- These segments are added using ones complement arithmetic in such a way that the total is also n-bits long
- The total is then complemented and appended to the end of the original data unit as redundancy bits, called checksum field.

At Receiver side

- The unit is divided into k sections, each of n bits.
- All sections are added together using one’s complement to get the sum.
- The sum is complemented.
- If the result is zero, the data are accepted: otherwise, they are rejected.



Performance

- ➢ The checksum detects all errors involving an odd number of bits.
- ➢ It detects most errors involving an even number of bits.
- ➢ If one or more bits of a segment are damaged and the corresponding bit or bits of opposite value in a second segment are also damaged, the sums of those columns will not change and the receiver will not detect a problem.

ERROR CORRECTION

It can be handled in two ways:

1) Error correction by retransmission.

2) Forward error correction

1) Error correction by retransmission: when an error is discovered, receiver can have the sender retransmit the entire data unit.

2) ERROR CORRECTING CODES

Concept of error-correction can be easily understood by examining the simplest case of single-bit errors. As we have already seen that a single-bit error can be detected by addition of a parity bit with the data, which needed to be send.

A single additional bit can detect error, but it's not sufficient enough to correct that error too. For correcting an error one has to know the exact position of error, i.e. exactly which bit is in error (to locate the invalid bits).

For example, to correct a single-bit error in an ASCII character, the error correction must determine which one of the seven bits is in error. To this, we have to add some additional redundant bits.

To calculate the numbers of redundant bits (r) required to correct d data bits, let us find out the relationship between the two. So we have (d+r) as the total number of bits, which are to be transmitted; then r must be able to indicate at least d+r+1 different values. Of these, one value means no error, and remaining d+r values indicate error location of error in each of d+r locations. So, d+r+1 states must be distinguishable by r bits, and r bits can indicates 2r states. Hence, 2r must be greater than d+r+1.

$$\mathbf{2^r >= d+r+1}$$

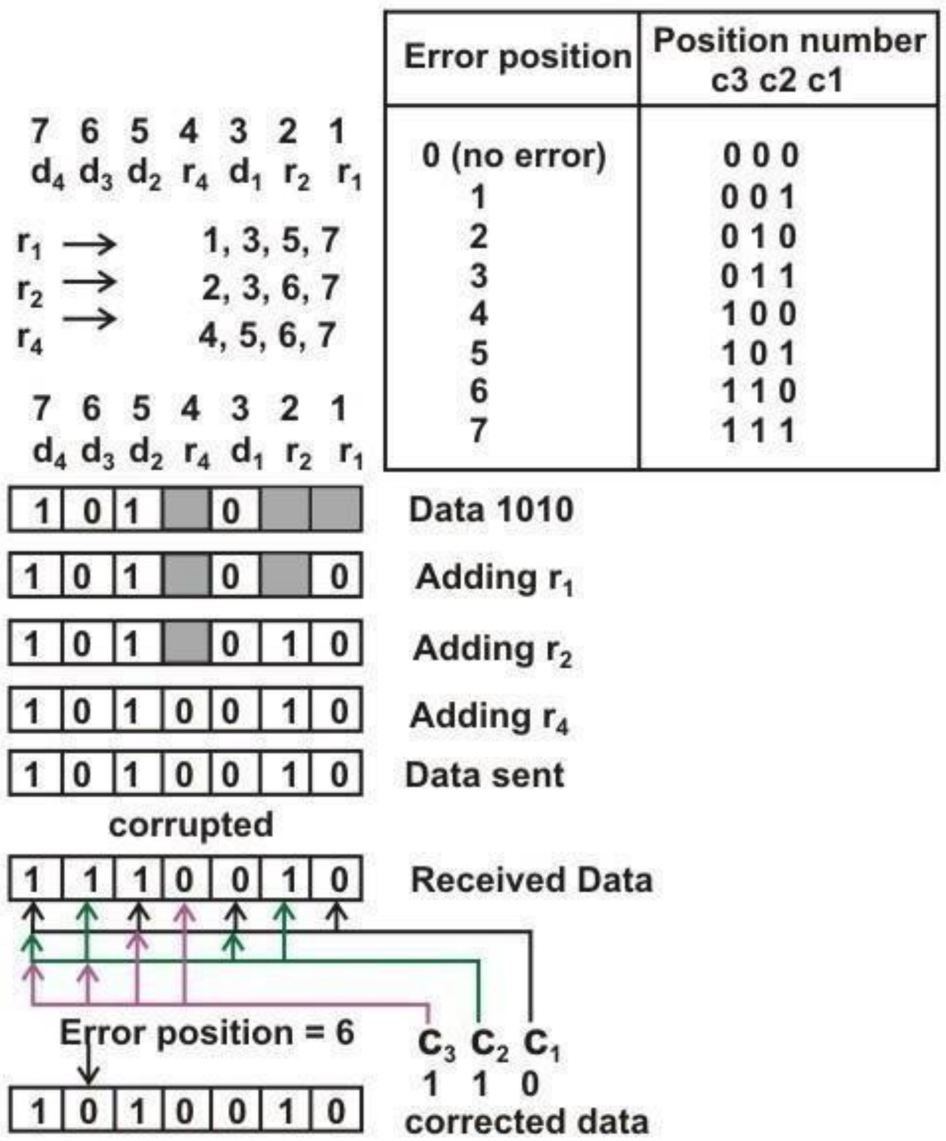
The value of r must be determined by putting in the value of d in the relation. For example, if d is 7, then the smallest value of r that satisfies the above relation is 4. So the total bits, which are to be transmitted is 11 bits (d+r = 7+4 =11).

Now let us examine how we can manipulate these bits to discover which bit is in error. A technique developed by R.W. Hamming provides a practical solution. The solution or coding scheme he developed is commonly known as **Hamming Code**.

Hamming code can be applied to data units of any length and uses the relationship between the data bits and redundant bits as discussed.

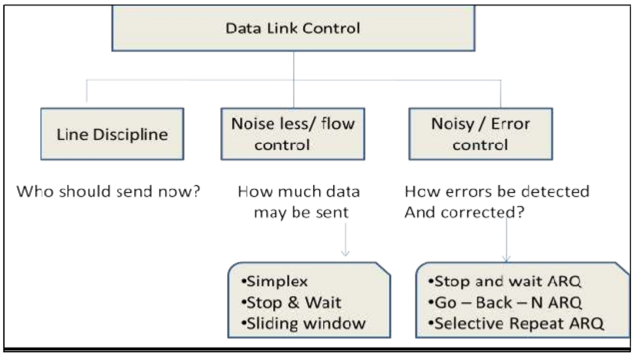
11	10	9	8	7	6	5	4	3	2	1
D	D	D	R	d	d	d	r	d	r	r

Positions of redundancy bits in hamming code



Error position	Position number c3 c2 c1
0 (no error)	0 0 0
1	0 0 1
2	0 1 0
3	0 1 1
4	1 0 0
5	1 0 1
6	1 1 0
7	1 1 1

ELEMENTARY DATA LINK PROTOCOLS



Noiseless channel:

A set of procedures used to restrict the amount of data, the sender can send before waiting the ACK.

i. **Simplest / Unrestricted simplex protocol: nickname-utopia:**

.Here no flow control

.No error control

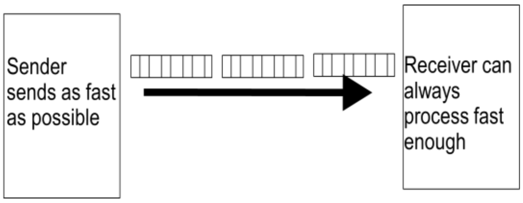
.It is unidirectional protocol.

.No sequence numbers and Acks are used here

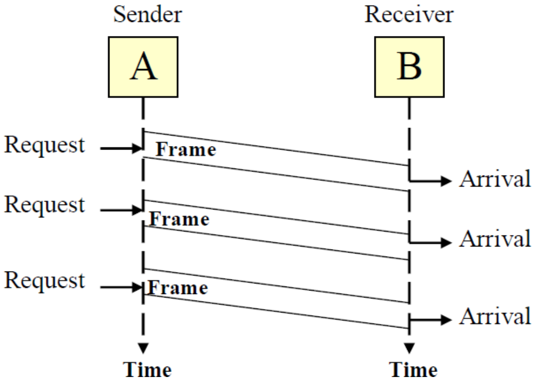
.Both sender & receiver network layers are always ready

.Infinite buffer space is available.

.Frames are never damaged or lost.


Sender sends as fast as possible
Receiver can always process fast enough

Flow diagram



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
    }
}
```

ii. Stop - & - Wait Protocol:

To prevent the receiver from becoming over whelmed with frames, we need to tell the sender to slow down, so there must be a feedback from the receiver to sender.

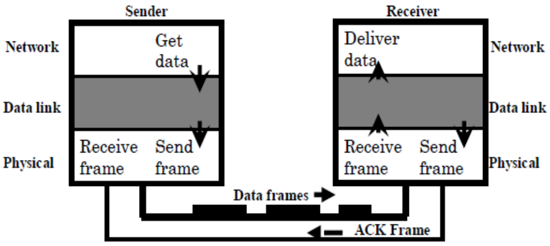
➢ Here the sender sends one frame and waits for an ack before sending the next frame.

➢ This process of alternately sending and waiting repeats until the sender transmits an End of Transmission (EOT) frame.

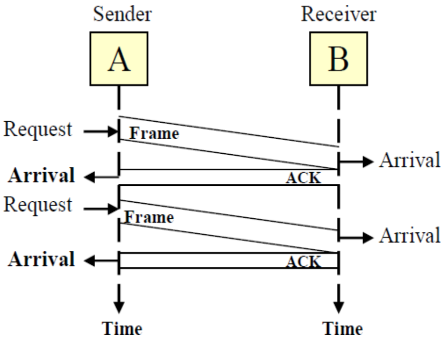
Assumptions in this protocol are

- 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.

Design:



Flow diagram:



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

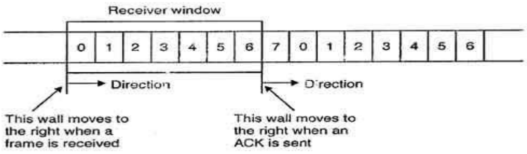
```
while(true)                    //repeat forever
cansend = true                //allow the first frame to go

  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
    }
}
```

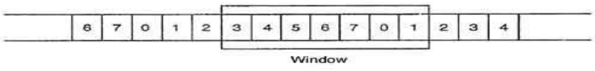
iii. Sliding Window Protocol:

In this sliding window method of flow control, several frames cab be in transit at a time.


Receiver window
0
1
2
3
4
5
6
7
0
1
2
3
4
5
6
Direction
Direction
This wall moves to the right when a frame is received
This wall moves to the right when an ACK is sent

Sliding window on receiver side

- The sliding window refers to imaginary boxes at both the sender and the receiver.
- Here as many frames can be in transit, they need the sequence numbers. In each window position, some of these sequence numbers define the frames that have been sent
- Frames may be acknowledged by receiver at any point even when window is not full on receiver side
- Frames may be transmitted by source even when window is not yet full on sender side.
- Therefore, a maximum of n-l frames may be sent before an acknowledgment
- Sequence number of transmitted frame is maintained in the header of each frame. If ***m*** is the number of ***bits*** in a frame, the sequence numbers range from 0 to $2^m - 1$. Ex: if m is 3, the only sequence numbers are 0 to 7. However, we can repeat the sequence, i.e 0,1,2,3,4,5,6,7,0,1,2…
- So the frames are numbered 0 to n-1, and the size of the window is n-1. Ex: if n=8 -> 0 to 7, max window size is n-1=7.
- To identify how many frames are transmitted and received, can be by size of the window.
- When receiver sends an ack, it includes the number of the next frame it ex ts to receive.
- The window can hold n-1 frames at either end; therefore, a max of n-1 frames may be sent before an Ack is required.



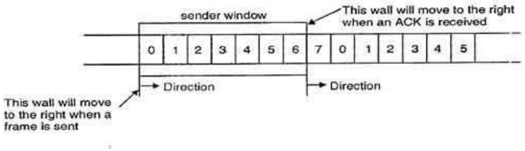
Sliding Window

Sender Window:

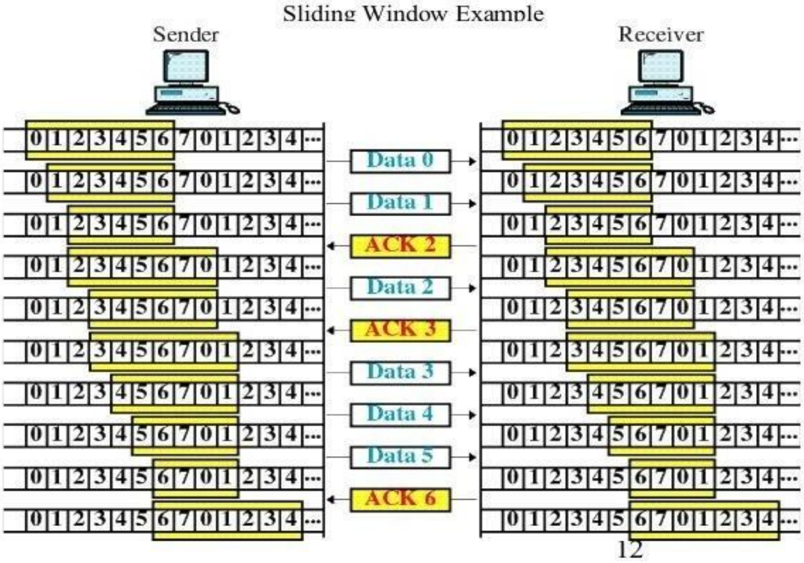
- At the beginning of a transmission, the sender's window contains n-l frames.
- As the frames are sent by source, the left boundary of the window moves inward, shrinking the size of window.
- When the receiver sends an ACK, the source's window expand i.e. (right boundary moves outward) to allow in a number of new frames equal to the number of frames acknowledged by that ACK.

Receiver Window:

- At the beginning of transmission, the receiver's window contains n-1 spaces for frame but not the frames
- As the new frames come in, the size of window shrinks from the left wall
- Therefore the receiver window represents not the number of frames received but the number of frames that may still be received without an acknowledgment ACK must be sent
- As soon as acknowledgment is sent, window expands to right, to include the number of frames equal to the number of frames acknowledged
- The window expands to include a number of new frame spaces equal to the number of the most recently acknowledged frame minus the number of previously acknowledged frame. For e.g., If window size is 7 and if prior ACK was for frame 2 & the current ACK is for frame 5 the window expands by three (5-2)

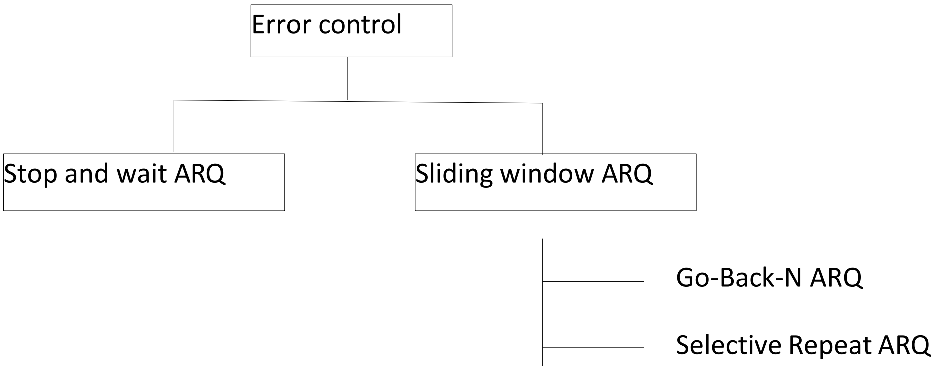

sender window
This wall will move to the right when an ACK is received
0
1
2
3
4
5
6
7
0
1
2
3
4
5
Direction
Direction
This wall will move to the right when a frame is sent

(a) Sliding Window with Size = 7



Noisy Channel: (Error control)

In the data link layer, the term error control refers primarily to methods of error detection and retransmission.



Noisy channel Assumptions are:

- Data transfer is unidirectional, but half-duplex link.
- 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.

Automatic Repeat Request (ARQ) :

Error control in data link layer is based on Automatic Repeat Request(ARQ), which means retransmission of data in three cases:

℗ Damaged Frame

℗ Lost Frame

℗ Lost Ack

ARQ is implemented in the data link layer as an addition to flow control. In fact the stop-and-wait flow control is usually implemented as stop-and-wait ARQ and sliding window is implemented as two variants of sliding window ARQ, called Go-back-n or selective-repeat ARQ.

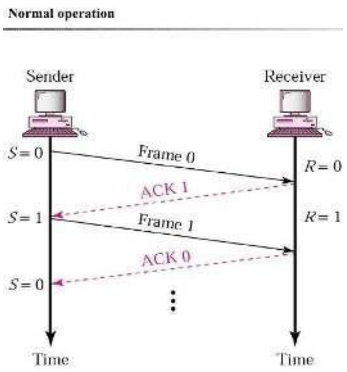
Stop-And-Wait ARQ:

In Stop-And-Wait ARQ protocol, first sender sends one frame and wait for an ack. When frame arrived at the receiver, then the receiver checks that is there any error in the frame or damage of the frame and sends ack accordingly.

Adds a simple error control mechanism to the Stop-&-Wait protocol.

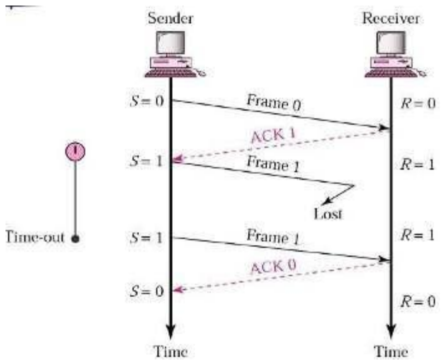
- Sending device keeps a copy of the last frame transmitted until acknowledgement for that frame is received.
- For Identification, data frames and ACK are numbered alternatively 0 and 1, to avoid duplication of frames.
- If an error discovered in a frame, indicating that it has been corrupted in transit, a NAK is returned.
- ACK frame uses Acknowledgement number, & always announces the sequence number
- of the next frame ex ted.
- Timers are used in case of loss of ACK frames.
- The sender has a control variable, S holds the number of the recently sent frame (0 or 1), and the Receiver has a control variable, R holds the number of the next frame ex ted(0 or 1).

Normal Operation:



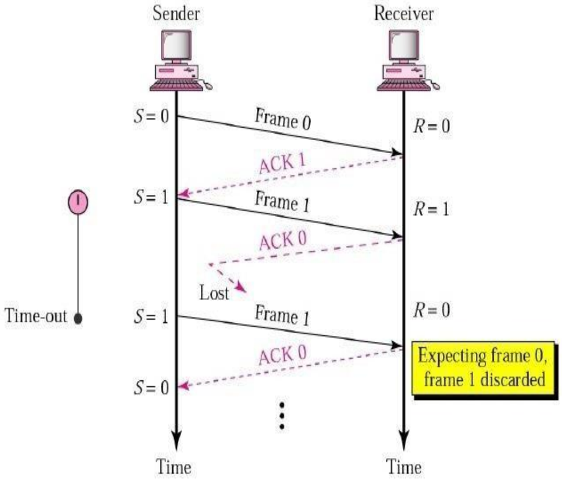
LOST FRAME:

when a receiver receives a *damaged* frame or , found frame is *lost*, then receiver don't send any ack. So after the timer at the sender expires, and another copy of previous frame is retransmitted.



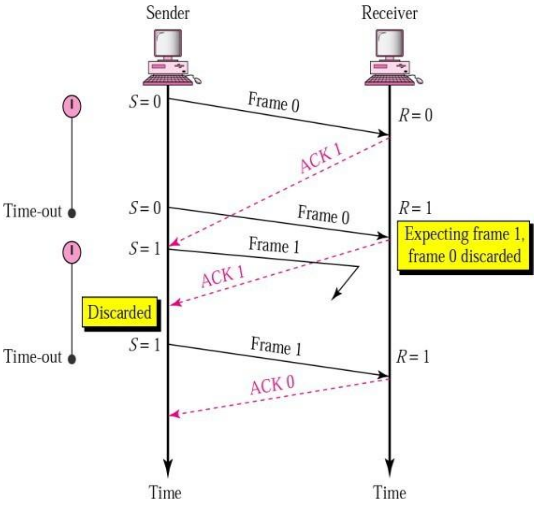
LOST ACK FRAME:

When ACK is lost or damaged, the sender does not have any idea about the frame whether it is receiver or not. So after timer expires for previous frame, the sender retransmits the same frame. As the receiver is already having that frame, it just discards the second copy of frame.



STOP-AND-WAIT, DELAYED ACK FRAME:

- The ACK can be delayed due to some channel problem at the receiver
- It is received after the timer has expired.
- But the Sender retransmitted a copy of frame 0. However, R =1 means receiver ex ts to see frame 1. Receiver discards the duplicate frame 0 and sends ACK.
- Sender receives 2 ACKs, it discards the second ACK.

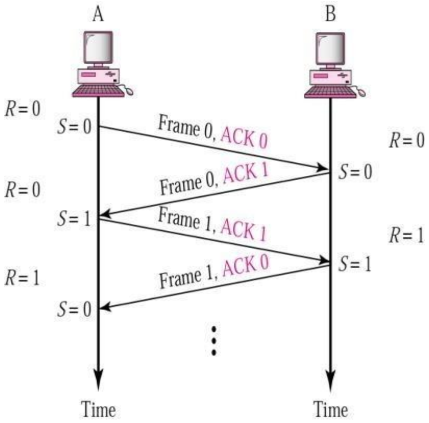


Bidirectional Transmission:

In stop-&-wait we had unidirectional transmission, we can have bi-directional transmission if the two parties have two separate channels for full duplex transmission or same channel for half- duplex transmission.

Piggybacking:

It is a method to combine a data frame with an acknowledgement. Piggybacking saves bandwidth.

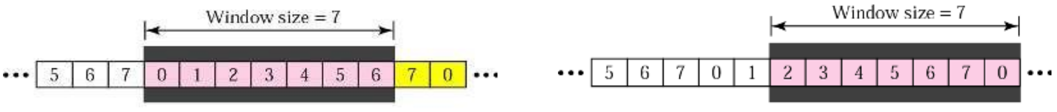


Go-Back-N ARQ:

- We can send up to W frames before worrying about ACKs.
- We keep a copy of these frames until the ACKs arrive.
- This procedure requires additional features to be added to Stop-and-Wait ARQ.
- Frames from a sender are numbered sequentially.
- We need to set a limit since we need to include the sequence number of each frame in the header.
- If the header of the frame allows m bits for sequence number, the sequence numbers range from 0 to $2^m - 1$. for m = 3, sequence numbers are: 0,1, 2, 3, 4, 5, 6, 7. However, we can repeat the sequence, i.e 0,1,2,3,4,5,6,7,0,1,2…

Sender Window:

- At the sending site, to hold the outstanding frames until they are acknowledged, we use the concept of a window.
- The size of the window is at most 2^m -1 where m is the number of bits for the sequence number.
- As the frames are sent by source, the left boundary of the window shrinks the size of window.
- When the receiver sends an ACK, the source's window expand to allow a number of new frames equal to the number of frames acknowledged by that ACK.

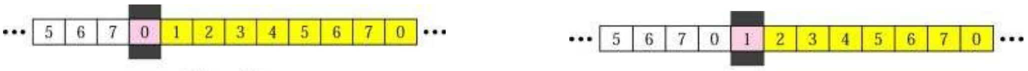


a. Before sliding

b. After sliding two frames

Receiver Window :

- Size of the window at the receiving site is always 1 in this protocol.
- Receiver is always looking for a s ific frame to arrive in a s ific order.
- Any frame arriving out of order is discarded and needs to be resent.

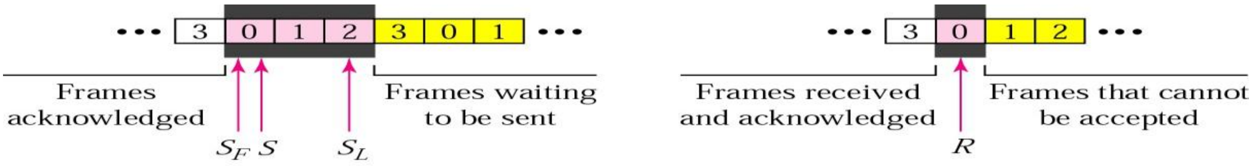


a. Before sliding

b. After sliding

Control Variables :

- Sender has 3 variables: S, S_F, and S_L
- S holds the sequence number of recently sent frame
- SF holds the sequence number of the first frame
- SL holds the sequence number of the last frame
- The size of the window is W, where W= S_F-S_L+1
- Receiver only has the one variable, R, that holds the sequence number of the frame it ex ts to receive. If the seq. no. is the same as the value of R, the frame is accepted, otherwise rejected.


... 3 0 1 2 3 0 1 ...
Frames acknowledged
Frames waiting to be sent
S_F S
S_L
... 3 0 1 2 ...
Frames received and acknowledged
Frames that cannot be accepted
R

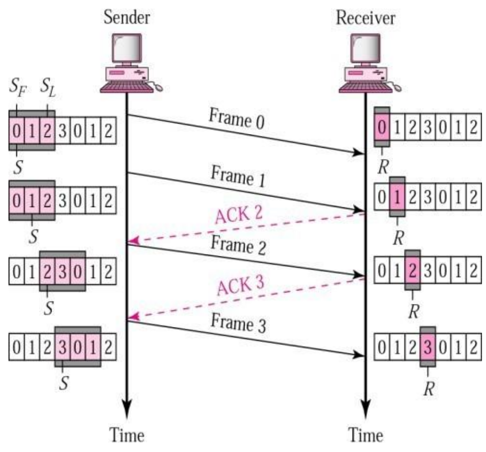
a. Sender window

b. Receiver window

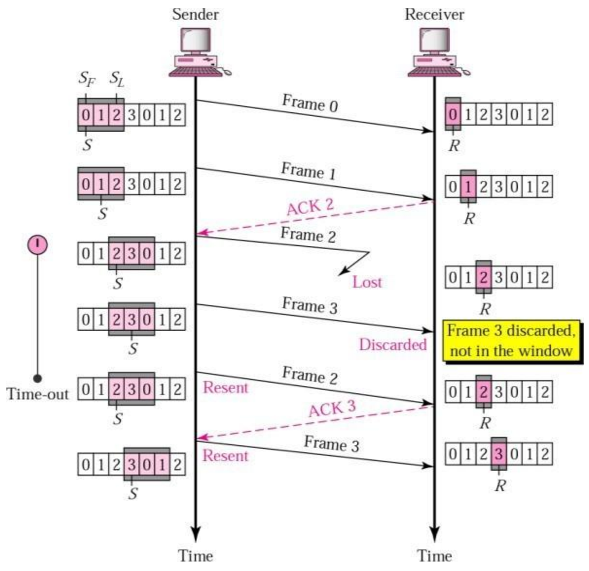
Acknowledgement

- Receiver sends positive ACK if a frame arrived safe and in order.
- If the frames are damaged/out of order, receiver is silent and discard all subsequent frames until it receives the one it is ex ting.
- The silence of the receiver causes the timer of the unacknowledged frame to expire.
- Then the sender resends 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 ACK for several frames.

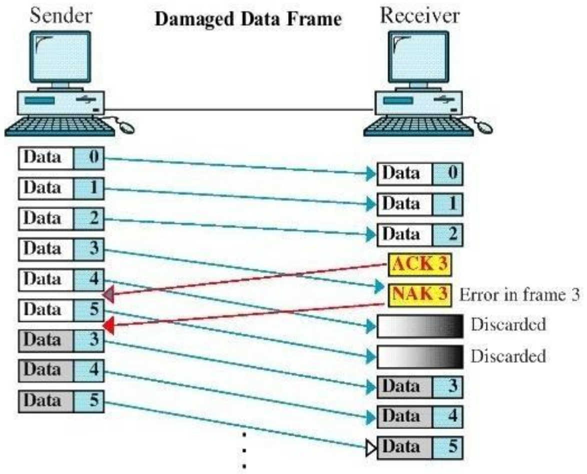
Go-Back-N ARQ, Normal Operation: **Go-Back-N ARQ, lost frame**

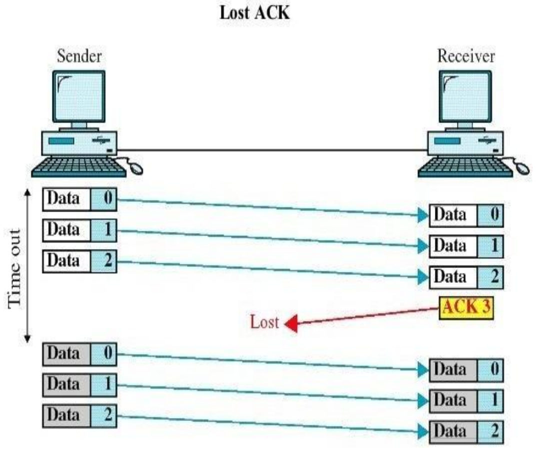


Go-Back-N ARQ, Damaged frame



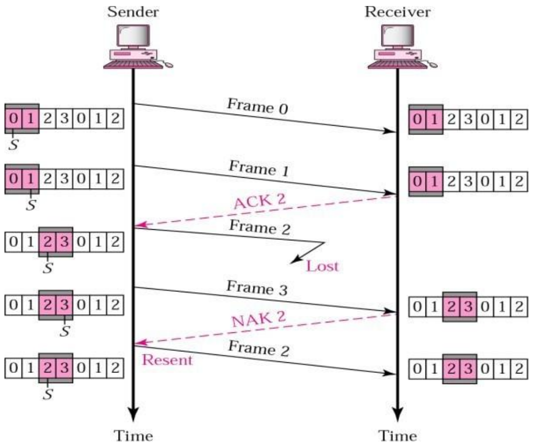
Go-Back-N ARQ, Damaged frame

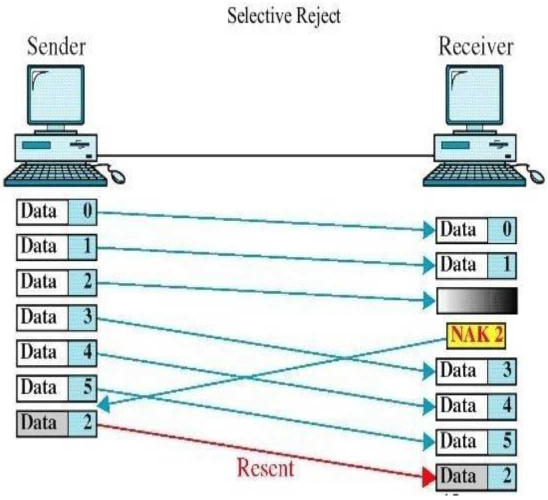




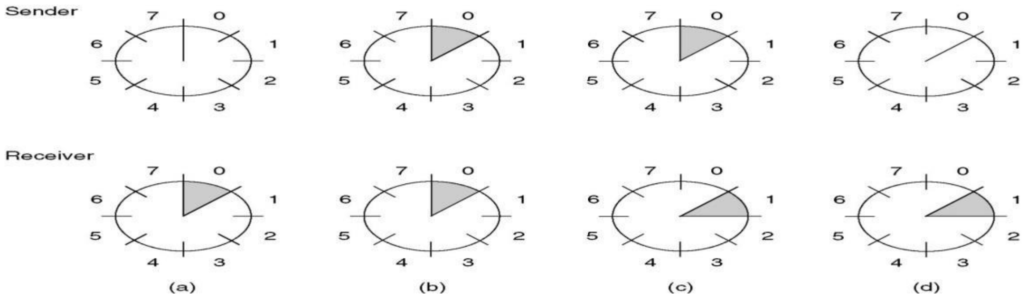
SELECTIVE REPEAT:

- Go-Back-N ARQ simplifies the process at the receiver site. Receiver only keeps track of only one variable, and there is no need to buffer out-of-order frames, they are simply discarded.
- However, Go-Back-N ARQ protocol is inefficient for noisy link. It is bandwidth inefficient and slows down the transmission.
- In Selective Repeat ARQ, only the damaged frame is resent. So more bandwidth efficient but more complex processing at receiver.
- It defines a negative ACK (NAK) to report the sequence number of a damaged frame before the timer expires.
- Size of the sender and receiver windows must be at most one-half of 2^{m}. If m = 2, window size should be $2^{m}/2 = 2$, or (**2^{m-1}**)





A One-Bit Sliding Window Protocol:



A sliding window of size 1, with a 3-bit sequence number.

(a) Initially.

(b) After the first frame has been sent.

(c) After the first frame has been received.

(d) After the first acknowledgement has been received.

Step1: The sender has to indicate the sequence number of transmitted frame

Step2: The sender must keep all the frames in its memory before sending the frame.

Step3: the receiver has to maintain a sequence no. of the ex ted frame.

Step4: The Data Link Layer fetches the packet from network layer and converts the packet into frames on the sender side.

Step5: The receiver data at receiver side is checked by data link layer.

Step6: If ex ted frame is arrived then the received window slided to forward direction

PIPELINING:

In networking, a task is often begun before the previous task has ended, is known as **pipelining.** There is no pipelining in Stop-and-wait ARQ because, sent frame is acknowledged before the next frame to be sent. Still pipelining apply to Go-Back-N ARQ and Selective Repeat ARQ because several frames can be sent before we receive news about previous frames
Pipelining improves the efficiency of transmission, and bandwidth efficiency.

QUESTIONS:

1. Data link protocols almost always put the CRC in a trailer, rather than in a header. Why?

   To compute the CRC, the entire data frame is shifted through a hardware register as the data passes by. The CRC is ready only when the last bit passes through and is appended at the end of the frame. Since the CRC is computed as the data is passing, it cannot be placed at the header of the frame because it is not yet ready.

A Simplex Protocol for a Noisy Channel:

Protocols in which the sender waits for a positive acknowledgement before transmitting the next frame which is often called PAR (Positive Acknowledgement with Retransmission). If the ACK is not arrived with in the time, then sender retransmit the same frame. ARQ (Automatic Repeat reQuest).

HDLC:

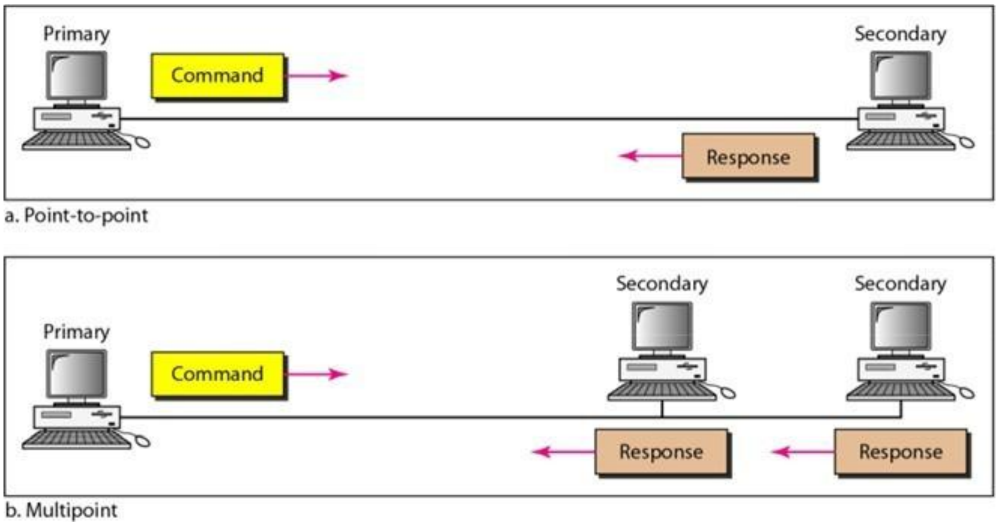
High-level Data Link Control (HDLC) is a bit-oriented protocol for communication over point-to-point and multipoint links. It implements the ARQ mechanisms

Configurations and Transfer Modes

HDLC provides two common transfer modes that can be used in different configurations: normal response mode (NRM) and asynchronous balanced mode (ABM).

Normal Response Mode

In normal response mode (NRM), the station configuration is unbalanced. We have one primary station and multiple secondary stations. A primary station can send commands; a secondary station can only respond. The NRM is used for both point-to-point and multiple-point links, as shown in below Figure.

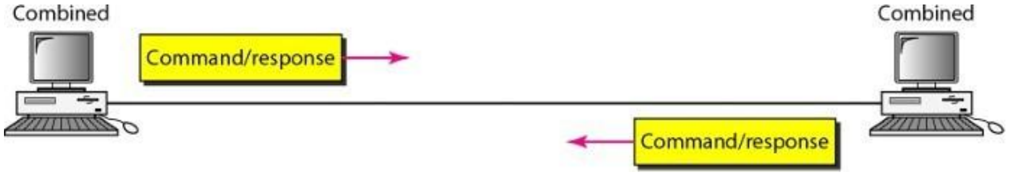


a. Point-to-point

b. Multipoint

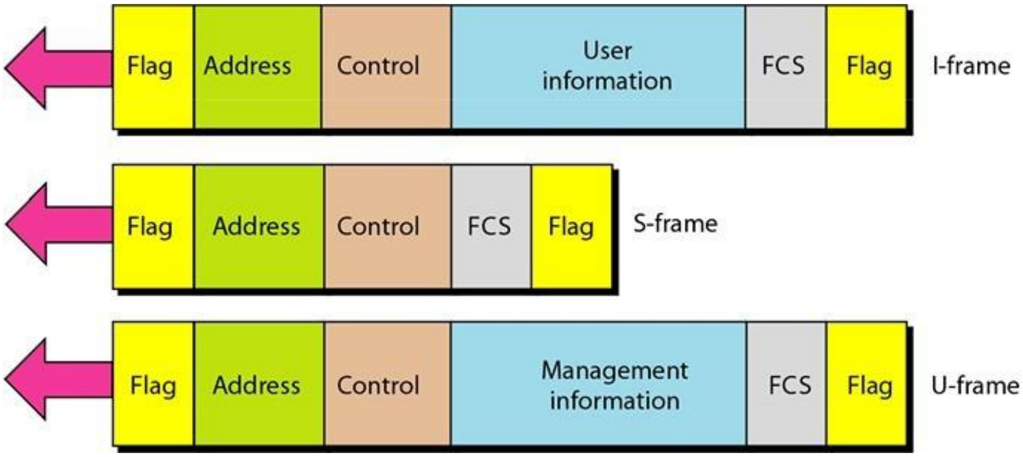
Asynchronous Balanced Mode

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), as shown in below Figure. This is the common mode today.



Frames

To provide the flexibility necessary to support all the options possible in the modes and configurations just described, HDLC defines three types of frames: information frames (I-frames), supervisory frames (S- frames), and unnumbered frames (V-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. V-frames are reserved for system management. Information carried by V-frames is intended for managing the link itself.



Fields

Let us now discuss the fields and their use in different frame types.

Flag field. The flag field of an HDLC frame is an 8-bit sequence with the bit pattern 01111110 that identifiesboth the beginning and the end of a frame and serves as a synchronization pattern for the receiver.

Address field. The second field of an HDLC frame contains the address of the secondary station. If a primary station created the frame, it contains a*to* address. If a secondary creates the frame, it contains *afrom* address. An address field can be 1 byte or several bytes long, depending on the needs of the network. One byte can identify upto 128 stations (lbit is used for another purpose).Larger networks require multiple-by the 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.

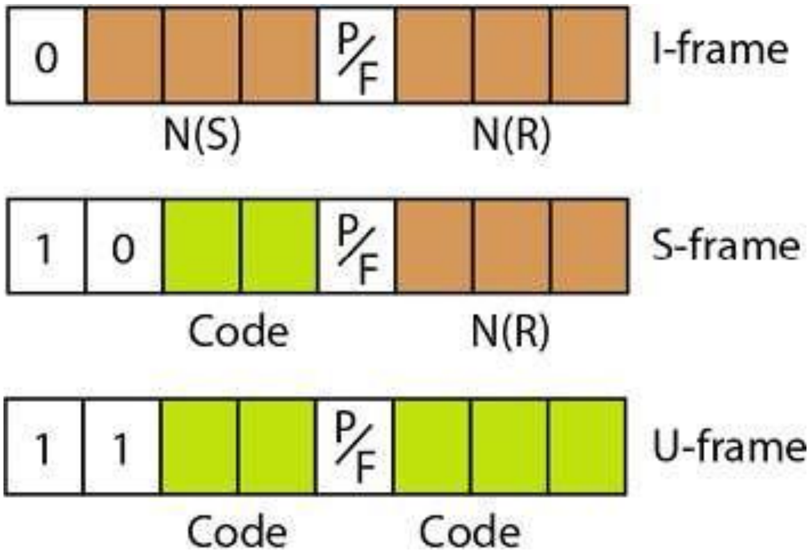
Control field. The control field is a 1- or 2-byte segment of the frame used for flow and error control. The interpretation of bits in this field depends on the frame type. We discuss this field later and describe its format for each frame type.

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. So let us discuss the format of this field in greater detail. The format is s ific for the type of frame, as shown in below Figure.



.

Control Field for I-Frames

I-frames are designed to carry user data from the network layer. In addition, they can include flow and error control information (piggybacking). The subfields in the control field are used to define these functions. The first bit defines the type. If the first bit of the control field is 0, this means the frame is an I-frame. The next 3 bits, called *N(S),* define the sequence number of the frame. Note that with 3 bits, we can define a sequence number between 0 and 7; but in the extension format, in which the control field is 2 bytes, this field is larger. The last 3 bits, called *N(R),* correspond to the acknowledgment number when piggybacking is used. The single bit between *N(S)* and *N(R)*is called the *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).

Control Field for S-Frames

Supervisory frames are used for flow and error control whenever piggybacking is either impossible or inappropriate (e.g., when the station either has no data of its own to send or needs to send a command or response other than an acknowledgment). S-frames do not have information fields. If the first 2 bits of the control field is 10, this means the frame is an S-frame. The last 3 bits, called *N(R),* corresponds to the acknowledgment number (ACK) or negative acknowledgment number (NAK) depending on the type of S- frame. The 2 bits called code is used to define the type of S-frame itself. With 2 bits, we can have four types of S-frames, as described below:

Receive ready (RR). If the value of the code subfield is 00, it is an RR S-frame. This kind of frame acknowledges the receipt of a safe and sound frame or group of frames. In this case, the value *N(R)* field defines the acknowledgment number.

Receive not ready (RNR). If the value of the code subfield is 10, it is an RNR S-frame. This kind of frame is an RR frame with additional functions. It acknowledges the receipt of a frame or group of frames, and it announces that the receiver is busy and cannot receive more frames. It acts as a kind of congestion control mechanism by asking the sender to slow down. The value of *NCR)* is the acknowledgment number.

Reject (REJ). If the value of the code subfield is 01, it is a REJ S-frame. This is a NAK frame, but not like the one used for Selective Repeat ARQ. It is a NAK that can be used in *Go-Back-N* ARQ to improve the efficiency of the process by informing the sender, before the sender time expires, that the last frame is lost or damaged. The value of *NCR)* is the negative acknowledgment number.

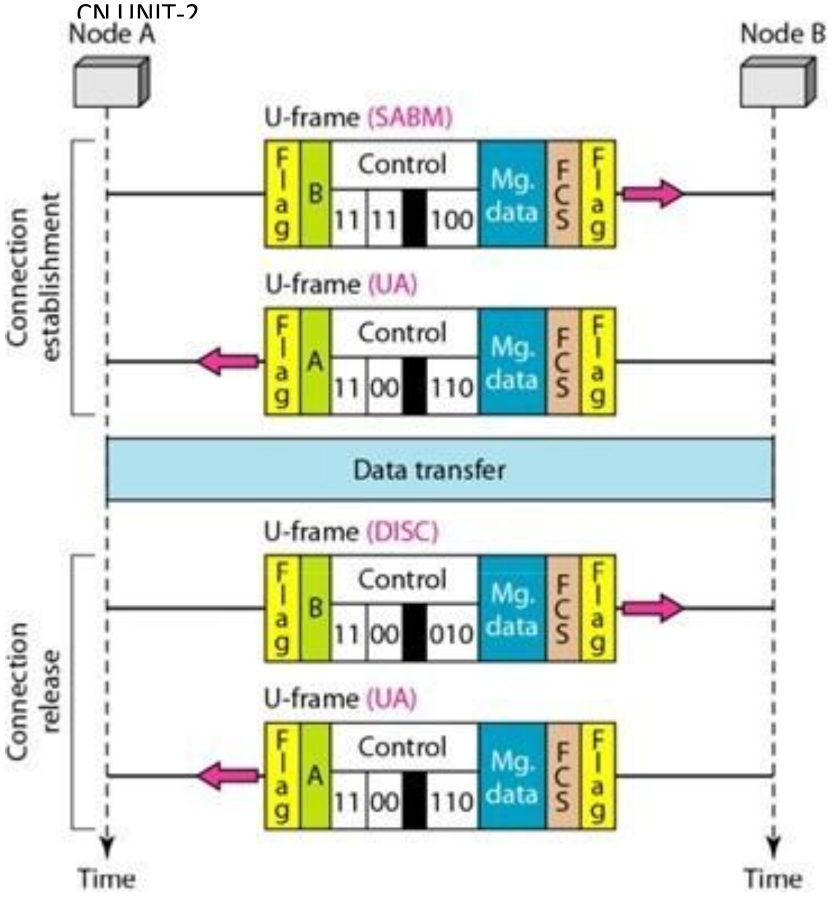
Selective reject (SREJ). If the value of the code subfield is 11, it is an SREJ S-frame. This is a NAK frame used in Selective Repeat ARQ. Note that the HDLC Protocol uses the term *selective reject* instead of *selective repeat.* The value of *N(R)* is the negative acknowledgment number.

Control Field for U-Frames

Unnumbered frames are used to exchange session management and control information between connected devices. Unlike S-frames, U-frames contain an information field, but one used for system management information, not user data. As with S-frames, however, much of the information carried by U- frames is contained in codes included in the control field. U-frame codes are divided into two sections: a 2-bit prefix before the P/F bit and a3-bit suffix after the P/F bit. Together, these two segments (5 bits) can be used to create up to 32 different types of U- frames.

Example: *Connection/Disconnection*

Below Figure shows how V-frames can be used for connection establishment and connection release. Node A asks for a connection with a set asynchronous balanced mode(SABM)frame; node B gives appositive response withanunnumberedacknowledgment(VA)frame.Afterthesetwoexchanges,datacanbetransferredbetween the two nodes (not shown in the figure).After data transfer, node A sends a DISC(disconnect)frame to release the connection; it is confirmed by node B responding with a VA (unnumbered acknowledgment).



POINT-TO-POINT PROTOCOL:

Although HDLC is a general protocol that can be used for both point-to-point and multipoint configurations, one of the most common protocols for point-to-point access is the Point-to-Point Protocol (PPP). 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. But to control and manage the transfer of data, there is a need for a point-to-point protocol at the data link layer. PPP is by far the most common.

PPP provides several services:

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

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

I. PPP does not provide flow control. A sender can send several frames one after another with no concern about overwhelming the receiver.

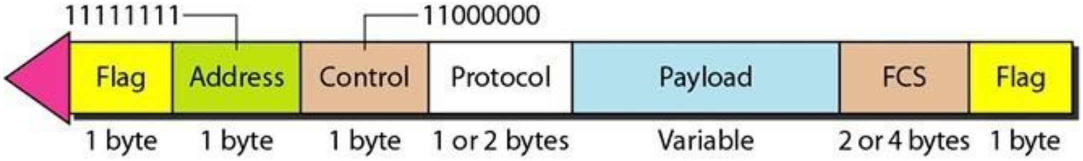
1. PPP has a very simple mechanism for error control. A CRC field is used to detect errors. If the frame is corrupted, it is silently discarded; the upper-layer protocol needs to take care of the problem. Lack of error control and sequence numbering may cause a packet to be received out of order.
2. PPP does not provide a sophisticated addressing mechanism to handle frames in a multi point configuration.

Framing:

PPP is a byte-oriented protocol. Framing is done according to the discussion of byte oriented protocols.

Frame Format

Below Figure shows the format of a PPP frame. The description of each field follows:



Flag. A PPP frame starts and ends with a I-byte flag with the bit pattern 01111110. Although this pattern is the same as that used in HDLC, there is a big difference. PPP is a byte-oriented protocol; HDLC is a bit-oriented protocol. The flag is treated as a byte, as we will explain later.

Address. The address field in this protocol is a constant value and set to 11111111(broadcast address). During negotiation (discussed later), the two parties may agree to omit this byte.

Control. This field is set to the constant value 11000000 (imitating unnumbered frames in HDLC). As we will discuss later, PPP does not provide any flow control. Error control is also limited to error detection. This means that this field is not needed at all, and again, the two parties can agree, during negotiation, to omit this byte.

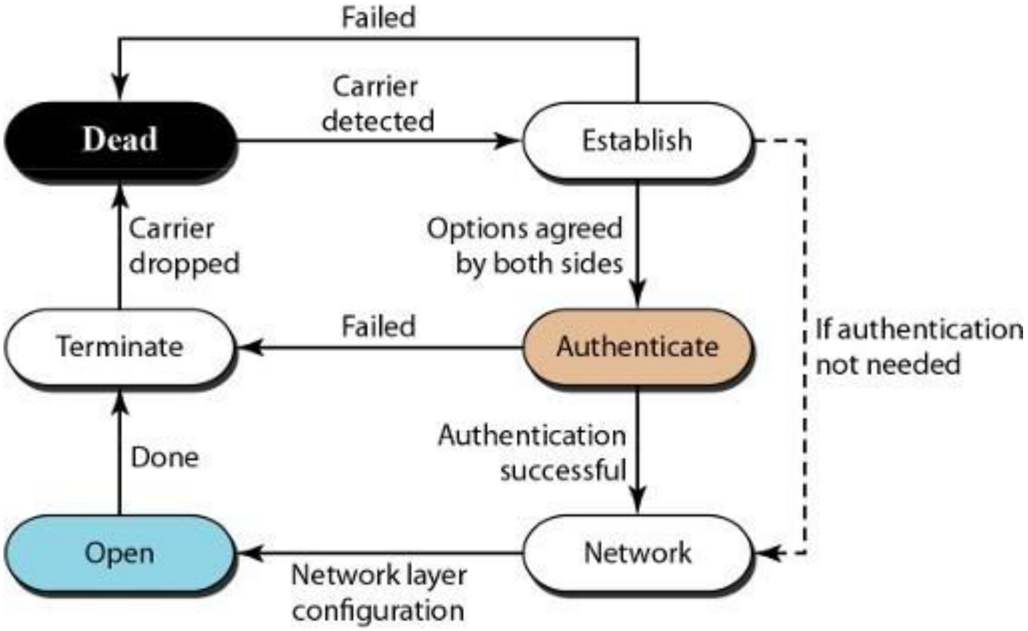
Protocol. The protocol field defines what is being carried in the data field: either user data or other information. We discuss this field in detail shortly. This field is by default 2 bytes long, but the two parties can agree to use only 1 byte.

Payload field. This field carries either the user data or other information that we will discuss shortly. The data field is a sequence of bytes with the default of a maximum of 1500 bytes; but this can be changed during negotiation. The data field is byte stuffed if the flag byte pattern appears in this field. Because there is no field defining the size of the data field, padding is needed if the size is less than the maximum default value or the maximum negotiated value.

FCS. The frame check sequence(FCS)is simply a 2-byteor4-bytestandard CRC.

Transition Phases

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


Failed
Carrier detected
Dead
Establish
Carrier dropped
Options agreed by both sides
Terminate
Failed
Authenticate
If authentication not needed
Done
Authentication successful
Open
Network
Network layer configuration

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 (ifauthenticationisrequired)ordirectlytothenetworkingphase.Thelinkcontrolprotocol packets, discussed shortly, are used for this purpose. Several packets may be exchanged here.

Authenticate. The authentication phase is optional; the two nodes may decide, during the establishment phase, not to skip this phase. However, if they decide to proceed with authentication, they send several authentication packets, discussed later. If the result is successful, the connection goes to the networking phase; otherwise, it goes to the termination phase.

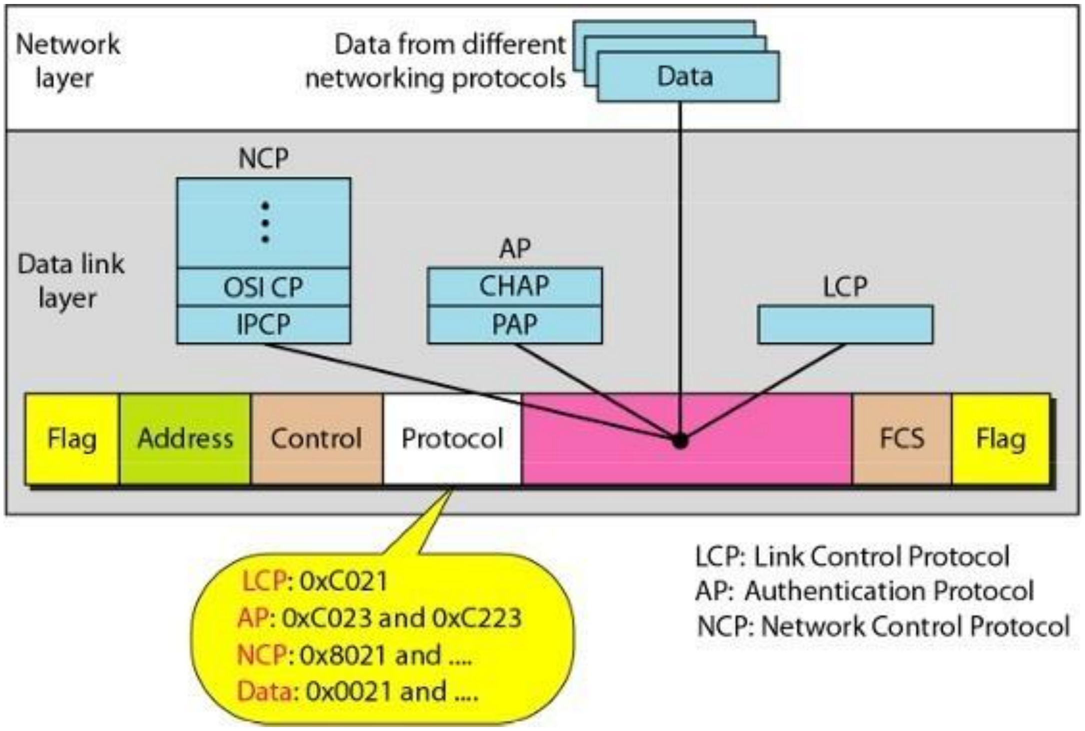
Network. In the network phase, negotiation for the network layer protocols takes place. PPP s ifies that two nodes establish a network layer agreement before data at the network layer can be exchanged.There as on is that PPP supports multiple protocols at the network layer. If a node is running multiple protocols simultaneously at the network layer, the receiving node needs to know which protocol will receive the data.

Open. In the open phase, data transfer takes place. When a connection reaches this phase, the exchange of data packets can be started. The connection remains in this phase until one of the endpoints wants to terminate the connection.

Terminate. In the termination phase the connection is terminated. Several packets are exchanged between the two ends for house cleaning and closing the link.

Multiplexing

Although PPP is a data link layer protocol, PPP uses another set of other protocols to establish the link, authenticate the parties involved, and carry the network layer data. Three sets of protocols are defined to make PPP powerful: the Link Control Protocol (LCP), two Authentication Protocols (APs), and several Network Control Protocols (NCPs). At any moment, a PPP packet can carry data from one of these protocols in its data field, as shown in below Figure. Note that there is one LCP, two APs, and several NCPs. Data may also come from several different network layers.

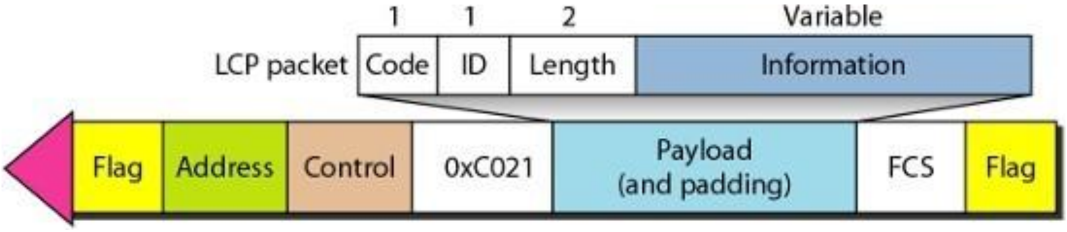


Link Control Protocol

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

All LCP packets are carried in the payload field of the PPP frame with the protocol field set to C021 in hexadecimal.

The code field defines the type of LCP packet.There are11types of packets as shown in below Table.



There are three categories of packets. The first category, comprising the first four packet types, is used for link configuration during the establish phase. The second category, comprising packet types 5 and 6, is used for link termination during the termination phase. The last five packets are used for link monitoring and debugging.

The ID field holds a value that matches a request with a reply. One endpoint inserts a value in this field, which will be copied into the reply packet. The length field defines the length of the entire LCP packet. The information field contains information, such as options, needed for some LCP packets.

There are many options that can be negotiated between the two endpoints. Options are inserted in the information field of the configuration packets. In this case, the information field is divided into three fields: option type, option length, and option data. We list some of the most common options in below Table.

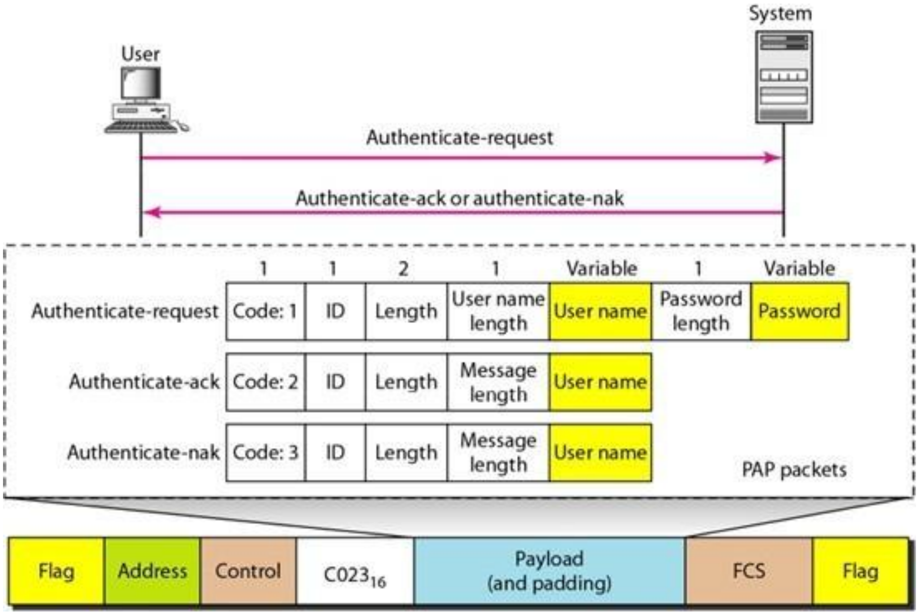
Authentication Protocols

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

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

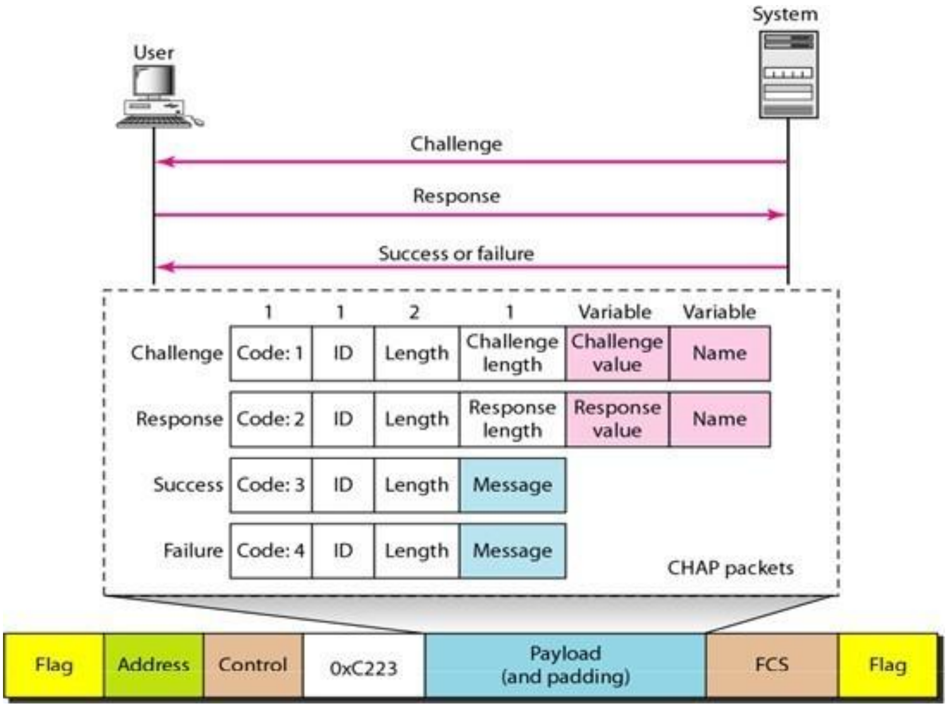
1. The user who wants to access a system sends an authentication identification (usually the user name) and a password.

2. The system checks the validity of the identification and password and either accepts or denies connection. Below Figure shows the three types of packets used by PAP and how they are actually exchanged. When a PPP frame is carrying any PAP packets, the value of the protocol field is OxC023. The three PAP packets are authenticate-request, authenticate-ack, and authenticate-nak. The first packet is used by the user to send the user name and password. The second is used by the system to allow access. The third is used by the system to deny access.



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

1. The system sends the user a challenge packet containing a challenge value, usually a few bytes.
2. The user applies a predefined function that takes the challenge value and the user's own password and creates a result. The user sends the result in the response packet to the system.
3. The system does the same. It applies the same function to the password of the user(known to the system)and the challenge value to create a result. If the result created is the same as the result sent in the response packet, access is granted; otherwise, it is denied. CHAP is more secure than PAP, es ially if the system continuously changes the challenge value. Even if the intruder learns the challenge value and the result, the password is still secret. Below Figure shows the packets and how they are used.

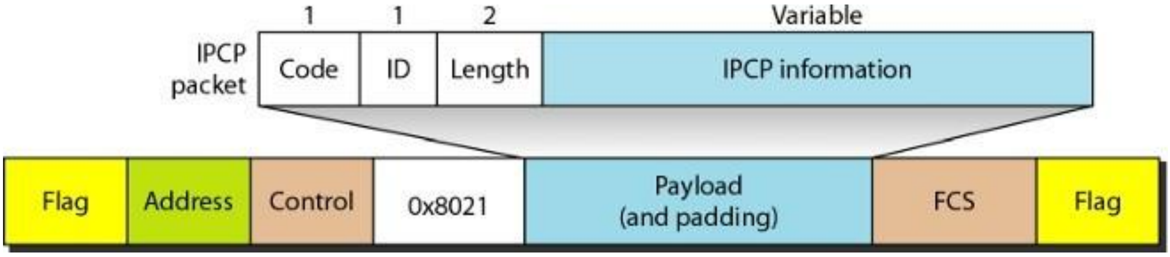


CHAP packets are encapsulated in the PPP frame with the protocol value C223 in hexadecimal. There are four CHAP packets: challenge, response, success, and failure. The first packet is used by the system to send the challenge value. The second is used by the user to return the result of the calculation. The third is used by the system to allow access to the system. The fourth is used by the system to deny access to the system.

Network Control Protocols

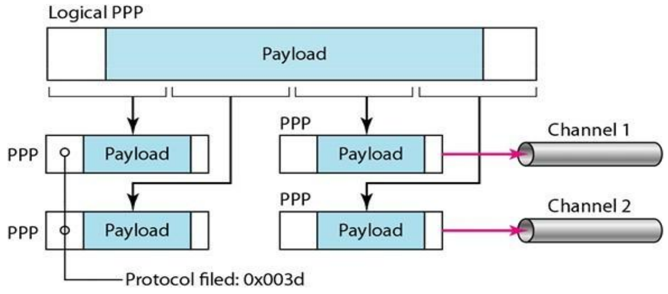
PPP is a multiple-network layer protocol. It can carry a network layer data packet from protocols defined by the Internet, OSI, Xerox, DECnet, AppleTalk, Novel, and so on. To do this, PPP has defined a s ific Network Control Protocol for each network protocol. For example, IPCP (Internet Protocol Control Protocol) configures the link for carrying IP data packets. Xerox CP does the same for the Xerox protocol data packets, and so on.

IPCP One NCP protocol is the Internet Protocol Control Protocol (IPCP). This protocol configures the link used to carry IP packets in the Internet. IPCP is es ially of interest to us.The format of an IPCP packet is shown in below Figure. Note that the value of the protocol field in hexadecimal is 8021.



Multilink PPP

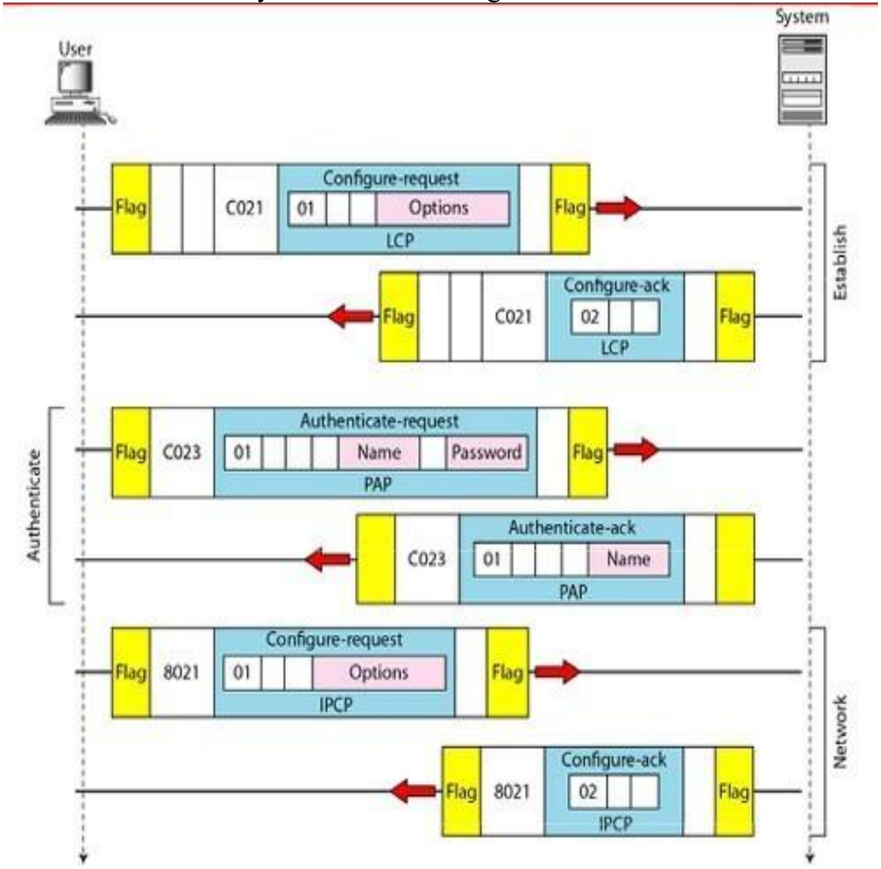
PPP was originally designed for a single-channel point-to-point physical link. The availability of multiple channels in a single point-to-point link motivated the development of Multilink PPP. In this case, a logical PPP frame is divided into several actual PPP frames. A segment of the logical frame is carried in the payload of an actual PPP frame, as shown in below Figure. To show that the actual PPP frame is carrying a fragment of a logical PPP frame, the protocol field is set to Ox003d. This new development adds complexity. For example, a sequence number needs to be added to the actual PPP frame to show a fragment's position in the logical frame.



Example

Let us go through the phases followed by a network layer packet as it is transmitted through a PPP connection. Below Figure shows the steps. For simplicity, we assume unidirectional movement of data from the user site to the system site (such as sending an e-mail through an ISP).

The first two frames show link establishment. We have chosen two options (not shown in the figure): using PAP for authentication and suppressing the address control fields. Frames 3 and 4 are for authentication. Frames 5 and 6 establish the network layer connection using IPCP.


System
User
Flag
C021
Configure-request
01
Options
LCP
Flag
Flag
C021
Configure-ack
02
LCP
Flag
Establish
Flag
C023
Authenticate-request
01
Name
Password
PAP
Flag
C023
Authenticate-ack
01
Name
PAP
Authenticate
Flag
8021
Configure-request
01
Options
IPCP
Flag
Flag
8021
Configure-ack
02
IPCP
Flag
Network