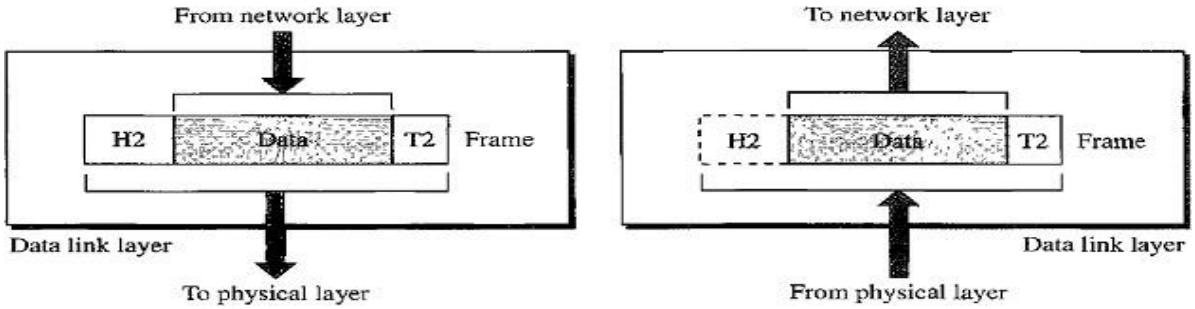
Unit-II

Data Link Layer

1. Explain in brief about the design issues in the data link layer.

The responsibility of **Data-Link Layer** is to transforming raw transmission facility into a **link** responsible for node-to-node communication (hop-to-hop communication).



Responsibilities of the Data Link Layer include:

1. Framing
2. Physical Addressing
3. Flow control
4. Error control
5. Media Access Control.

Framing

The data link layer divides the stream of bits received from the network layer into manageable data units called frames. In simple terms data link layer is responsible for moving frames from one node to another node.

Physical Addressing

The data link layer adds a header to the frame to define the addresses of the sender and receiver of the frame.

Flow Control

If the rate at which the data are absorbed by the receiver is less than the rate at which data are produced in the sender, the data link layer imposes a flow control mechanism to avoid overwhelming the receiver.

Error Control

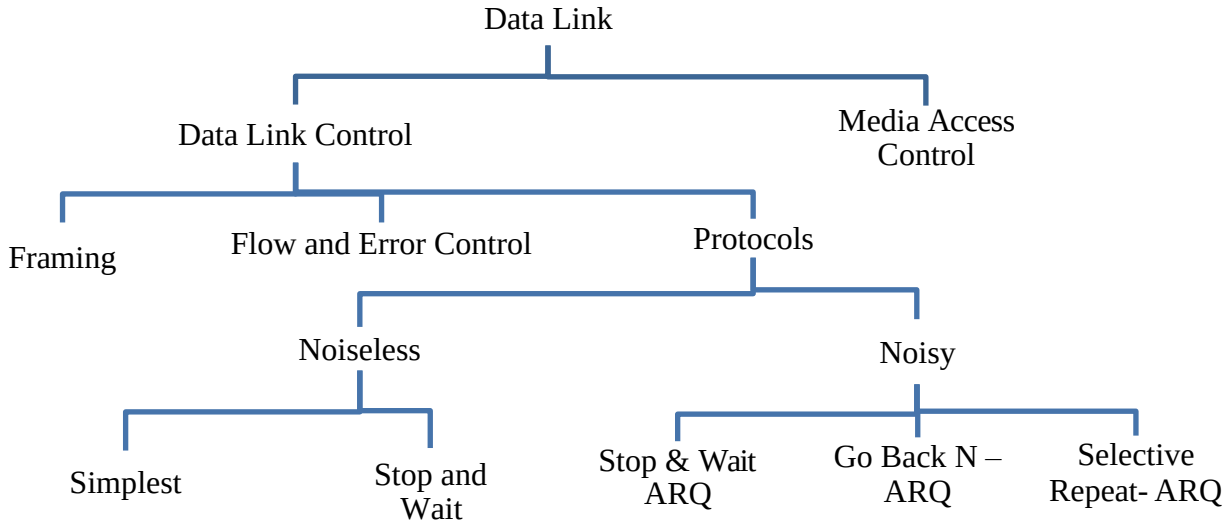
The data link layer also adds reliability to the physical layer by adding mechanisms to detect and retransmit damaged, duplicate, or lost frames.

Media Access Control

When two or more devices are connected to the same link, data link layer protocols are necessary to determine which device has control over the link at any given time.

FUNCTIONS OF DATA LINK LAYER

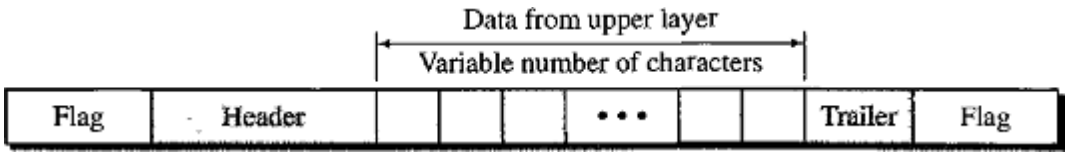
The functionality and sub functionalities of Data Link Layer are given below:



2. Explain in detail about bit stiffing and character stuffing ?

Character Stuffing / Byte Stuffing

In a character stuffing, data to be carried are 8-bit characters from a coding system such as ASCII. The Frame format in Character Stuffing is given below:



Character Stuffing uses: Header, Trailer and a Flag.

- **Header** carries the source and destination addresses and other control information.
- **Trailer** which carries error detection or error correction redundant bits, these are also multiples of 8 bits.
- To separate one frame from the next frame, an 8-bit (1-byte) **Flag** is added at the beginning and the end of a frame. The flag signals receiver either start or end of a frame.

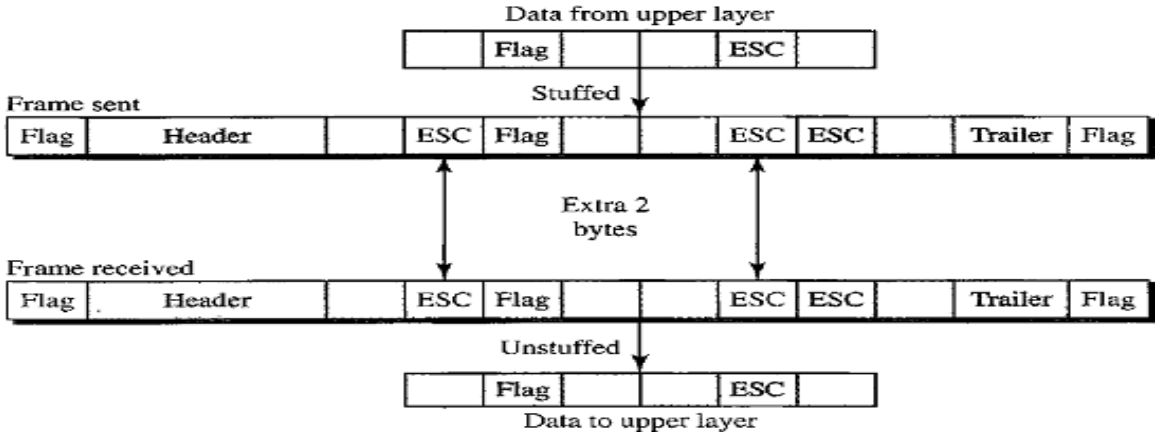
Disadvantages of Character Stuffing

- Character oriented framing is useful for text transfer not useful for audio video etc.
- Any pattern used for the flag could also be part of the information.
- If this happens, the receiver, when it encounters this pattern in the middle of the data, thinks it has reached the end of the frame and the treats the next bit as new frame.

To fix this problem a **Byte Stuffing** strategy is introduced.

- In byte stuffing a special byte is added to the data section of the frame when there is a character with the same pattern as the flag.
- The data section is stuffed with an extra byte called Escape character (ESC).
- Whenever the receiver encounters the ESC character, it removes it from the data section and treats the next character as data, not a delimiting flag.

Figure shows the byte stuffing and Unstuffing:



Problems with Byte Stuffing

- If the text contains one or more escape characters followed by a flag, the receiver removes the escape character, but keeps the flag, which is incorrectly interpreted as the end of the frame.

Solution

- To solve this problem, the escape characters that are part of the text must also be marked by another escape character.

Disadvantages of character / Byte stuffing Procedure

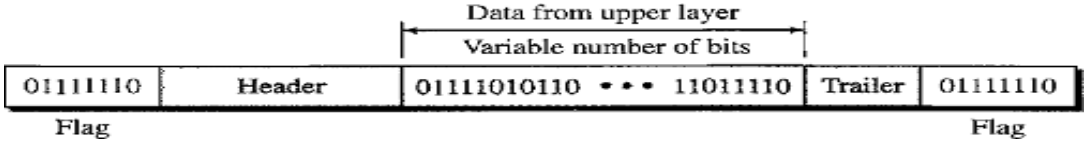
- The universal coding systems (Unicode) in use today have 16-bit and 32-bit characters that conflict with 8-bit characters.
- Character stuffing deals with 8-bit characters but todays systems using 16 bits, 32 bits and 64 bit characters hence there will be conflict.

The solution for this problem is using ***Bit Oriented Approach.***

<u>Bit stuffing</u>

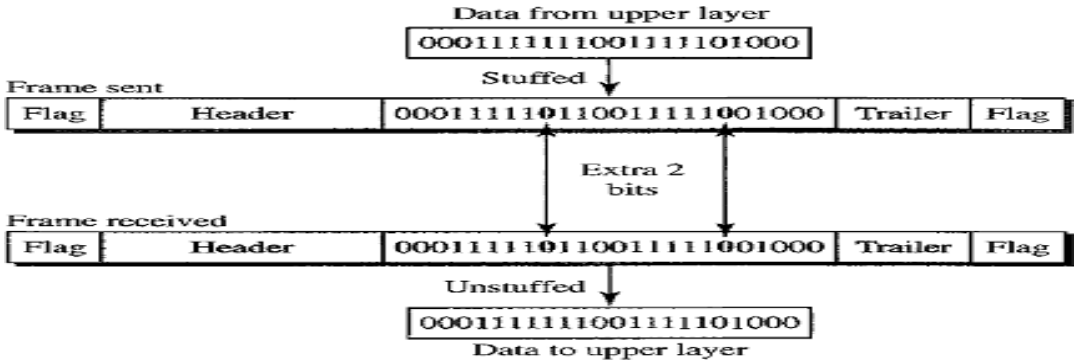
- It is used for text, graphic, audio, video, and so on.
- In bit stuffing the data section of a frame is a sequence of bits to be interpreted by the upper layer.
- In addition to header and trailer, we need a delimiter to separate one frame from other frame.
- Most protocols use a special 8-bit pattern flag 01111110 as the delimiter to define the beginning and the end of the frame.

Frame format in bit stuffing is give below figure:



- In bit stuffing, if a 0 and five consecutive 1- bits are encountered, an extra 0 is added.
- This extra stuffed bit is eventually removed from the data by the receiver.

Note: the extra bit is added after one 0 followed by five 1s regardless of the value of the next bit. (i.e.) when 0**11111**00 is a part of the data, then also we have to add "0" after five 1"s . Hence the data will be 0**11111**000



Advantages of Bit Stuffing

If the flag like pattern 01111110 appears in the data, it will change to 011111010 (stuffed) and is not mistaken as a flag by the receiver. The real flag 01111110 is not stuffed by the sender and is recognized by the receiver.

3. Explain the algorithm for CRC method of error checking.

Cyclic Codes

In a cyclic code, if a codeword is cyclically shifted (rotated), the result is another codeword. Example: If 1011000 is a codeword and we cyclically left-shift, then 0110001 is also a codeword.

Dataword	1011000
Codeword	0110001

Cyclic Redundancy Check (CRC)

CRC is used in networks such as LANs and WANs. We can create cyclic codes to correct errors. The above figure is a possible design for the encoder and decoder.

CRC Encoder

- **In** the encoder, the dataword has **k bits** and the codeword has **n bits**.
- The size of the dataword is augmented by adding **(n – k) number of 0's** to the right-hand side of the word.
- The **n-bit** result is fed into the generator.
- The generator uses a divisor of size **n - k + 1** predefined and agreed by both sender and reciever.
- The generator divides the augmented dataword by the divisor **(modulo-2 division)**.
- The quotient ofthe division is discarded;
- The remainder ($r_2 r_1 r_0$) is appended to the dataword to create the codeword.

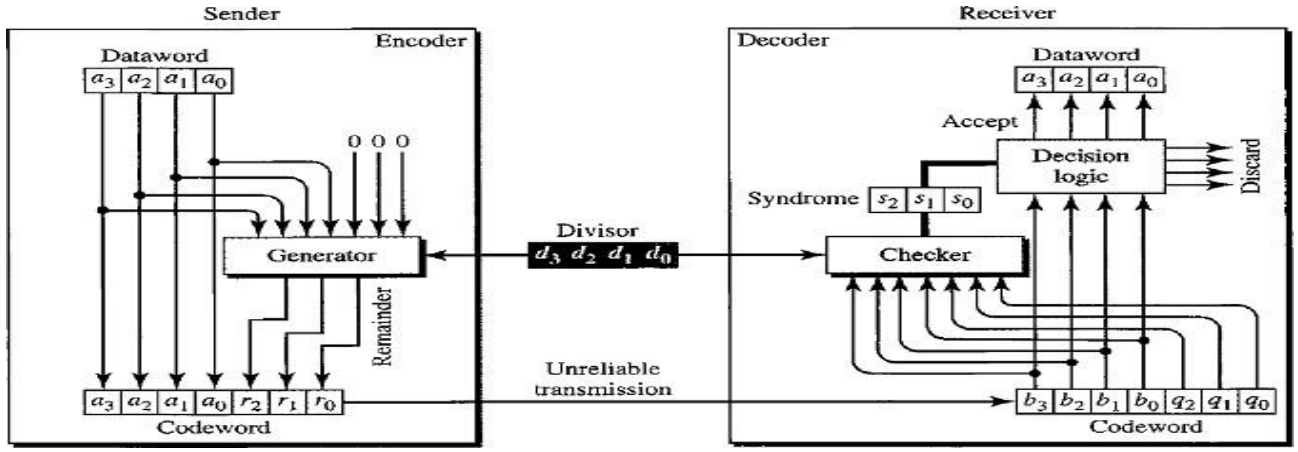
  Let us take

  k=4 bits

  n=7 bits

  Appended Dataword Size = **(n-k)** = 3.

  Divisor Size = **(n-k+1)** =4.



Decoder

- The decoder receives the possibly corrupted codeword.
- A copy of all *n* bits is fed to the checker which is a replica of the generator.
- The remainder produced by the checker is a syndrome of *n* - *k* (3 here) bits, which is fed to the decision logic analyzer. The analyzer has a simple function.
- If the syndrome bits are all 0‟s, the 4 leftmost bits of the codeword are accepted as the dataword (interpreted as no error); otherwise, the 4 bits are discarded (error).

Example: A CRC code with C(7, 4)

Dataword	*Codeword*	*Dataword*	*Codeword*
0000	0000000	1000	1000101
0001	0001011	1001	1001110
0010	0010110	1010	1010011
0011	0011101	1011	1011000
0100	0100111	1100	1100010
0101	0101100	1101	1101001
0110	0110001	1110	1110100
0111	0111010	1111	1111111

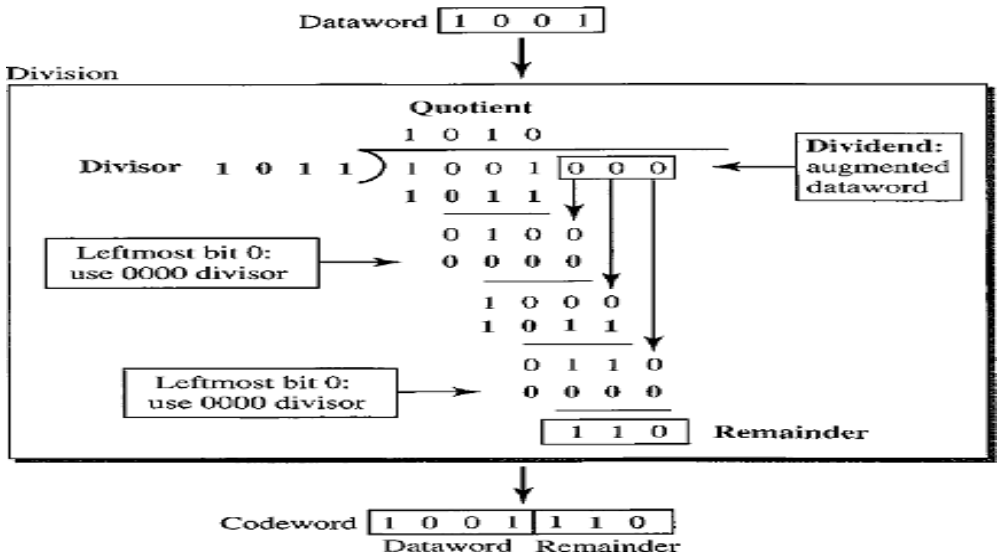
In the above table the dataword size is 4 and codeword size is 7. Codeword can be obtained by applying the CRC procedure as we mentioned above. Now let us check for the dataword 1001, and how we get codeword 1001110.

Encoder

The encoder takes the dataword and augments it with **(n – k)** number of 0‟s. It then divides the augmented dataword by the divisor. Let us take the divisor 1011.The value 1011 will agreed by both sender and receiver.

Note: We use XOR operation in the above division.

- As in decimal division, the process is done step by step.
- In each step, a copy of the divisor is XORed with the 4 bits of the dividend.
- The result of the XOR operation (remainder) is 3 bits is used for the next step after 1 extra bit is pulled down to make it 4 bits long.
- If the leftmost bit of the dividend is 0, the step cannot use the regular divisor; we need to use an all-0‟s divisor.
- When there are no bits left to pull down, we have a result.
- The 3-bit remainder forms the check bits ($\mathbf{r_2 r_1 r_0}$). They are appended to the dataword to create the codeword.

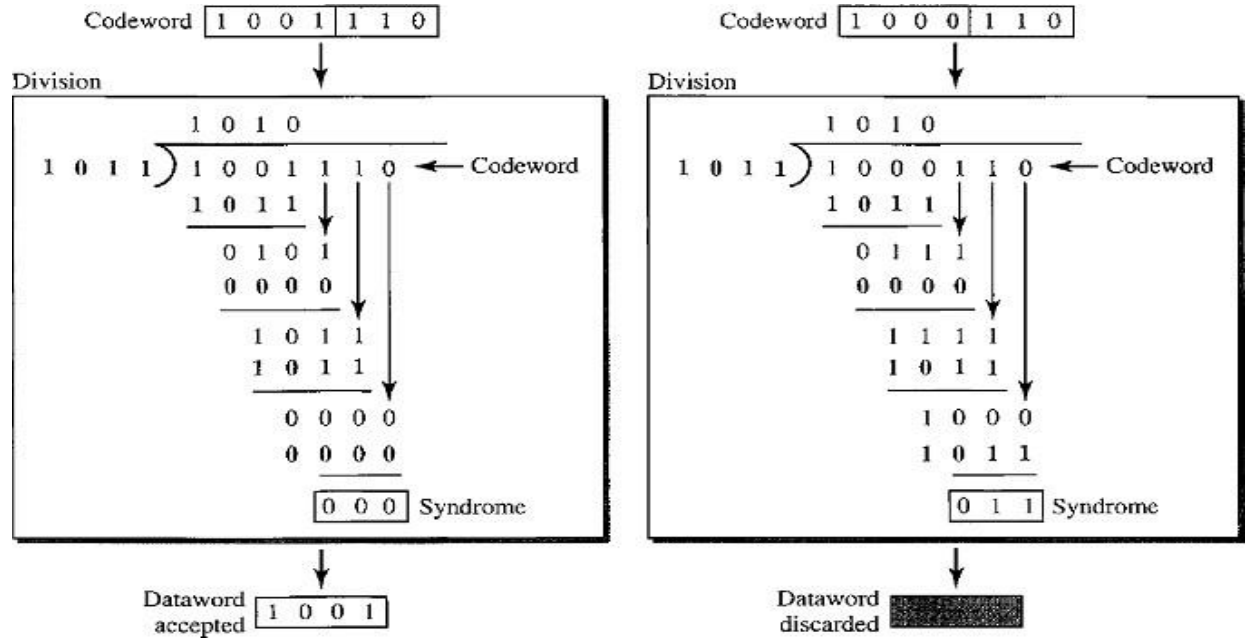


Decoder

- The codeword can change during transmission.
- The decoder does the same division process as the encoder.
- The remainder of the division is the syndrome.
- If the syndrome is all 0‟s, there is no error; the dataword is separated from the received codeword and accepted. Otherwise, everything is discarded.

The below figure shows two cases:

- The left-hand figure shows the value of syndrome when no error has occurred; the syndrome is 000.
- The right-hand part of the figure shows the case in which there is one single error. The syndrome is not all 0‟s (it is 011).



4. In detail, explain the various ALOHA protocols.

ALOHA

ALOHA was developed at the University of Hawaii in early **1970**. It was designed for a Wireless radio LAN, but it can be used on any shared medium. Due to shared medium there are potential collisions in this arrangement.

There are two types of methods in ALOHA

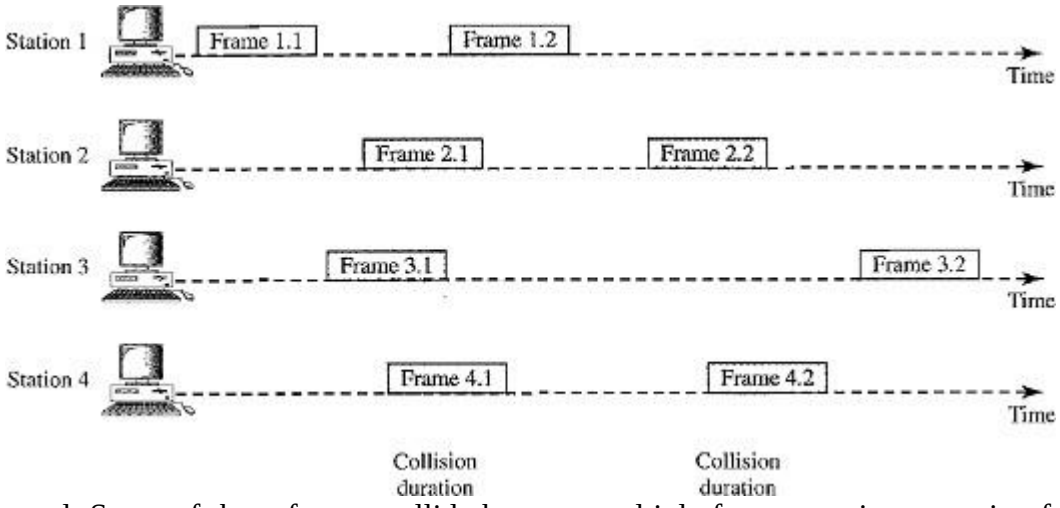
i. Pure ALOHA
ii. Slotted ALOHA

Pure ALOHA

The original ALOHA or Pure ALOHA is a simple protocol.

The idea is that each station sends a frame whenever it has a frame to send, there is only one channel to share, and there is the possibility of collision between frames from different stations.

- In the above figure, there are four stations each sending two frames and shares the same

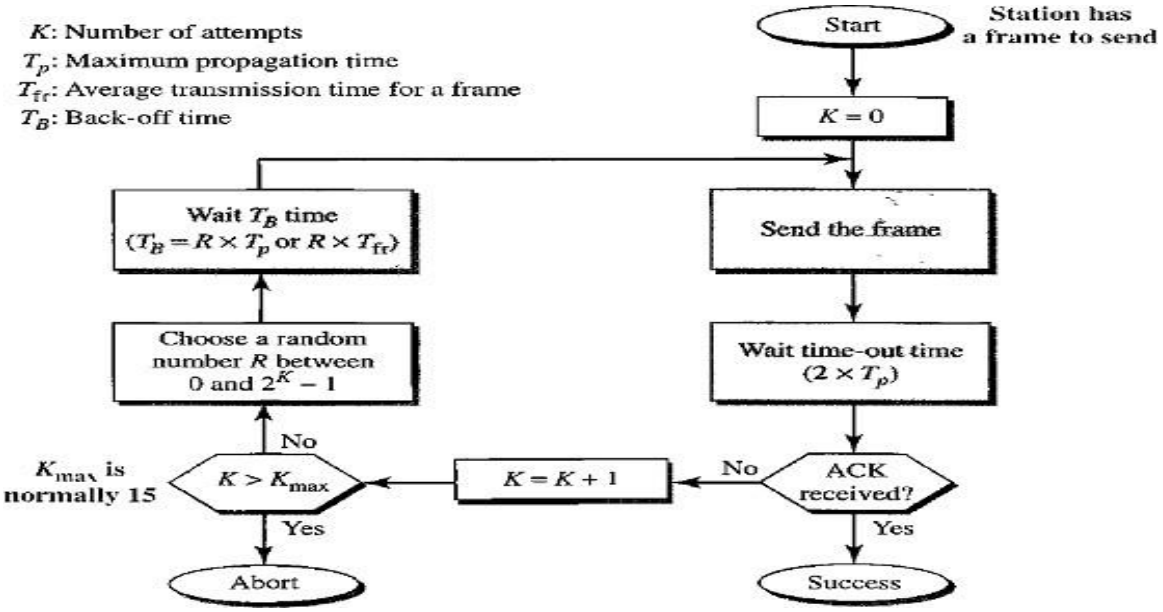


channel. Some of these frames collide because multiple frames are in contention for the shared channel.

- By observing the above figure only 2 frames frame 1.1 and frame 3.2 can be delivered at receiver, and the remaining frames collide with each other and they are lost or discarded at the receiver side.
- The pure ALOHA protocol relies on Acknowledgments (ACK) from the receiver. When a station sends a frame, it expects the receiver to send an acknowledgment.
- If the acknowledgment does not arrive after a time-out period, the station assumes that the frame (or the acknowledgment) has been destroyed and resends the frame.
- A collision involves two or more stations. If all these stations try to resend their frames after the time-out, the frames will collide again.

- Pure ALOHA dictates that when the time-out period passes, each station waits a random amount of time before resending its frame. The randomness will help avoid more collisions; this time is called the Back-Off Time $\mathbf{T_B}$.
- Pure ALOHA has a second method to prevent congesting the channel with retransmitted frames. After a maximum number of retransmission attempts $\mathbf{K_{max}}$ a station must give up and try later.

The procedure for pure ALOHA is given in the figure:



- The time-out period is equal to the maximum possible round-trip propagation delay, which is twice the amount of time required to send a frame between the two most widely separated stations (2 x T_p).
- The back-off time T_B is a random value that normally depends on K (the number of attempted unsuccessful transmissions).
- For each retransmission, a multiplier in the range 0 to $\boldsymbol{2^K - 1}$ is randomly chosen and multiplied by $\mathbf{T_p}$ (maximum propagation time) or $\boldsymbol{T_{fr}}$ (Frame transmission time or the average time required to send out a frame) to find $\mathbf{T_B}$.
- Note that in this procedure, the range of the random numbers increases after each collision. The value of $\mathbf{K_{max}}$ is usually chosen as **15**.

Vulnerable time

Vulnerable time is the length of time, in which there is a possibility of collision.

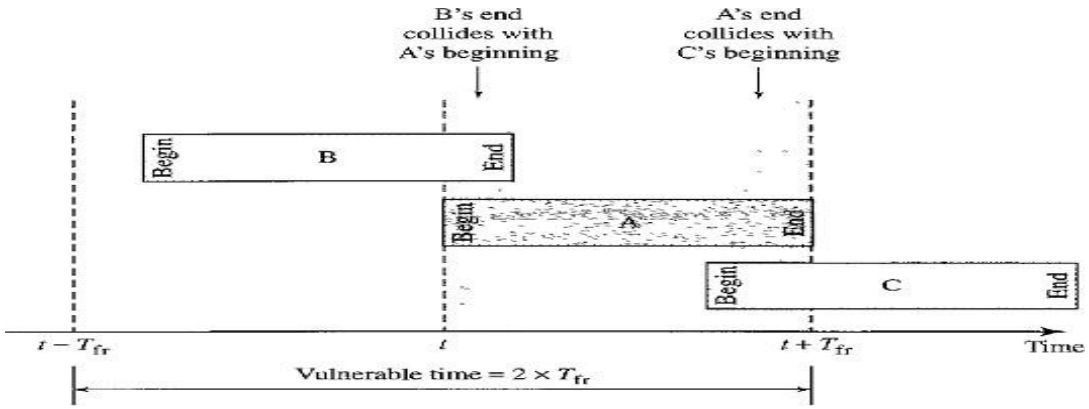
Let us assume that the stations send fixed-length frames with each frame taking $\mathbf{T_{fr}}$**'s** to send.

Station A sends a frame at time **t.**

Now station B has already sent a frame between **t** - $\mathbf{T_{fr}}$ and **t**.

This leads to a collision between the frames from station A and station B.

The end of B's frame collides with the beginning of A's frame.



Suppose that station C sends a frame between **t** and **t + T_{fr}** .

Here, there is a collision between frames from station A and station C. The beginning of C's frame collides with the end of A's frame.

The vulnerable time, during which a collision may occur in pure ALOHA, is 2 times the frame transmission time.

Pure ALOHA vulnerable time = **2 x T_{fr}**

Throughput

- The throughput for pure ALOHA is **$S = G \times e^{-2G}$**
- The maximum throughput **S_{max}** = 0.184 when G = (½).
- (i.e.) one frame is generated during two frame transmission times, then 18.4 percent of these frames reach their destination successfully.

<u>Slotted ALOHA</u>

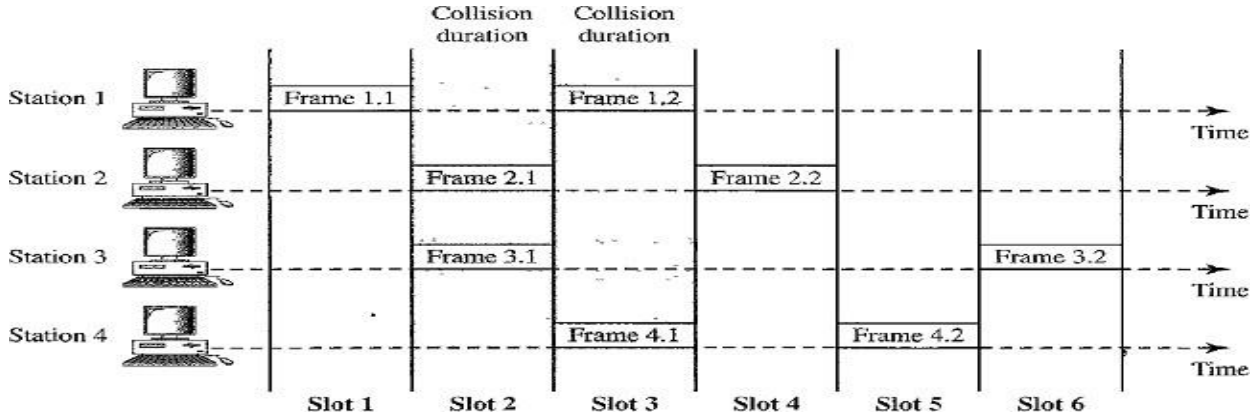
Pure ALOHA has a vulnerable time of **2 x T_{fr}** . This is so because there is no rule that defines when the station can send.

- A station may send soon after another station has started or soon before another station has finished.
- Slotted ALOHA was invented to improve the efficiency of pure ALOHA.
- In slotted ALOHA we divide the time into slots of **T_{fr}**'s and force the station to send only at the beginning of the time slot.
- A station is allowed to send only at the beginning of the synchronized time slot, if a station misses this moment, it must wait until the beginning of the next time slot.
- This means that the station which started at the beginning of this slot has already finished sending its frame.
- There is still the possibility of collision if two stations try to send at the beginning of the same time slot.

i.e. the vulnerable time for slotted ALOHA is one-half that of pure ALOHA.

Slotted ALOHA vulnerable time = **T_{fr}**.

Below figure shows an example of frame collisions in slotted ALOHA.



Throughput

- The throughput for slotted ALOHA is $S =: G \times e^{-G}$.
- The maximum throughput $S_{max} == 0.368$ when G =1.
- If a frame is generated during one frame transmission time, then 36.8 percent of these frames reach their destination successfully.

5. Explain about GBN Sliding Window Protocol.

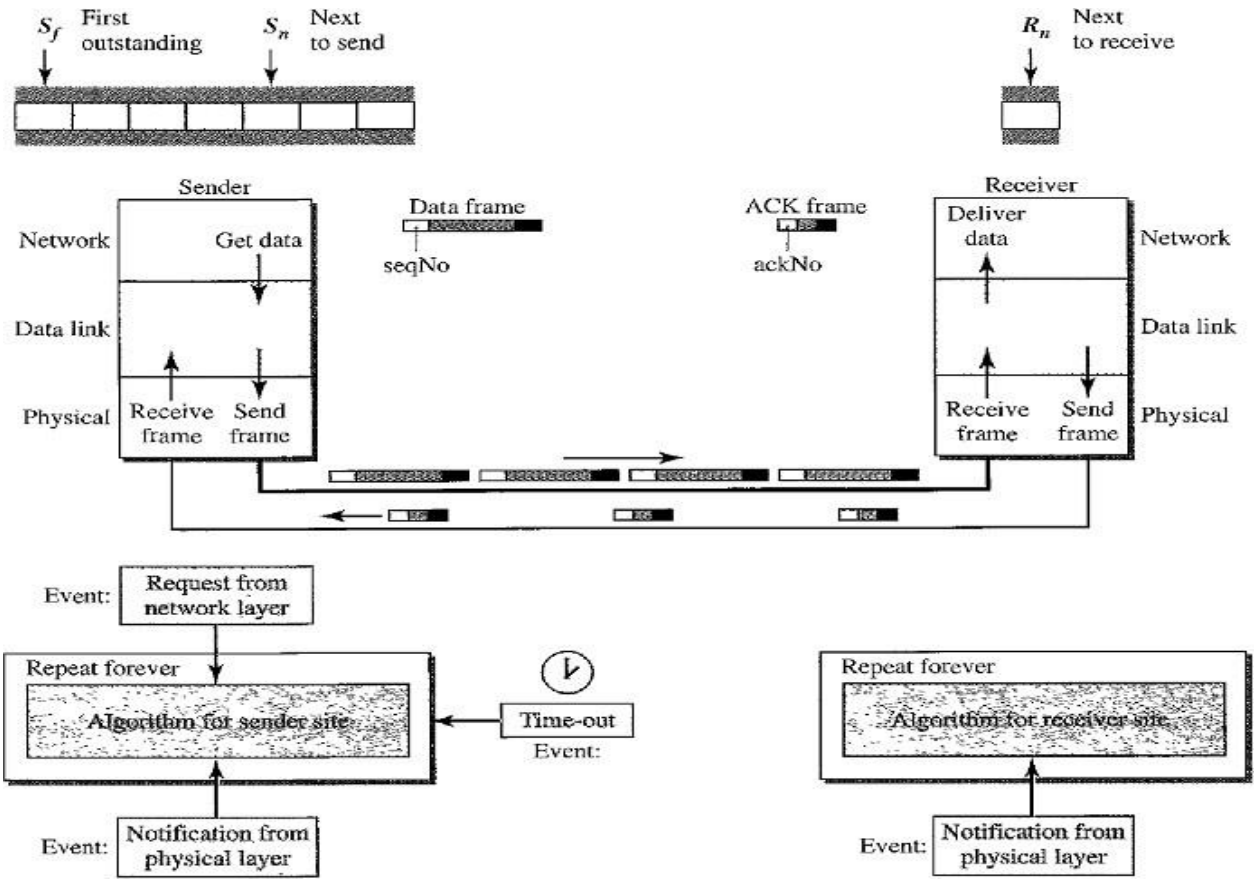
In this protocol

- Sender can send several frames before receiving acknowledgments.
- Sender keep a copy of these frames until the acknowledgments arrive.

Design of Go-Back-N ARQ

The idea is similar to Stop-and-Wait ARQ, multiple frames can be in transit in the forward direction, and multiple acknowledgments in the reverse direction.

The difference is that the send window allows us to have as many frames in transition as there are slots in the send window.



This protocol uses the following concepts:

1. Sequence Numbers
2. Sliding Window
3. Timers
4. Acknowledgement
5. Retransmission (Resending a Frame)

Sequence Numbers

- It is a number given to each frame included in the header.
- The header of the frame allows *m* bits for the sequence number, the sequence numbers range from 0 to 2^m-1.

- The sequence numbers are repeated after the number 2^m-1 (i.e) the sequence numbers are modulo-2^m

Example:
If m=4 , the sequence numbers are 0,1,2,3,4,5……14,15,0,1,2,3,4,5,6,……..

Sliding Window

Sliding Window is an abstract concept that defines the sender and receiver needs to deal with only part of the possible sequence numbers.

There are 2 types of windows are used:
i. Send Window
ii. Receive Window

The range that is the concern of sender is called the send sliding window or Send Window

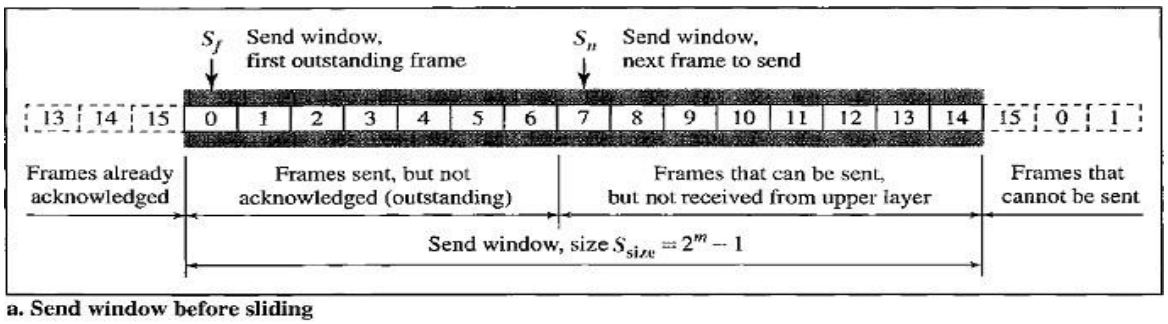
The range that is the concern of receiver is called receive sliding window or receive window.

Send Window

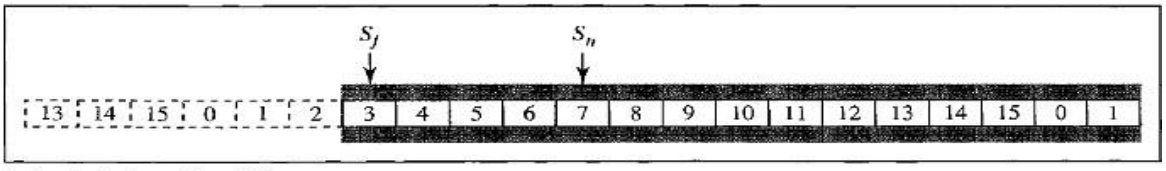
- The send window is an imaginary box covering the sequence numbers of the data frames which can be in transit.
- In each window position, some of these sequence numbers define the frames that have been sent; others define those that can be sent.
- The maximum size of the send window is **$2^m - 1$.**

Example: Let us take m=4. Size of window=15.

Consider the below figure:



a. Send window before sliding



b. Send window after sliding

The window at any time divides the possible sequence numbers into four regions.

1. The first region (the left side of the window) defines the sequence numbers belonging to frames that are already acknowledged. The sender don‟t need to keep copies of the frames.
2. The second region defines the range of sequence numbers belonging to the frames that are sent and have an unknown status. The sender needs to wait to find out if these frames have been received or were lost. These frames are called outstanding frames.

3. The third range defines the range of sequence numbers for frames that can be sent. The corresponding data packets have not yet been received from the network layer.
4. The fourth region defines sequence numbers that cannot be used until the window slides.

The window uses three variables define its size and location at any time.

i. S_f defines the sequence number of the first (oldest) outstanding frame.
ii. S_n holds the sequence number that will be assigned to the next frame to be sent.
iii. S_{size} defines the size of the window, which is fixed in our protocol.

The acknowledgments in this protocol are cumulative, meaning that more than one frame can be acknowledged by an ACK frame.

In the above figure frames 0, 1, and 2 are acknowledged, so the window has slid to the right three slots.

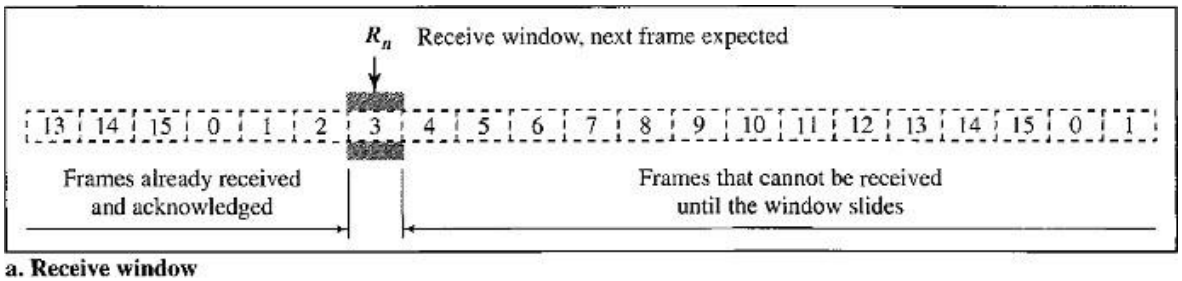
Receive Window

- The receive window makes sure that the correct data frames are received and that the correct acknowledgments are sent.
- The size of the receive window is always 1.
- The receiver is always looking for the arrival of a specific frame.
- Any frame arriving out of order is discarded and needs to be resent.

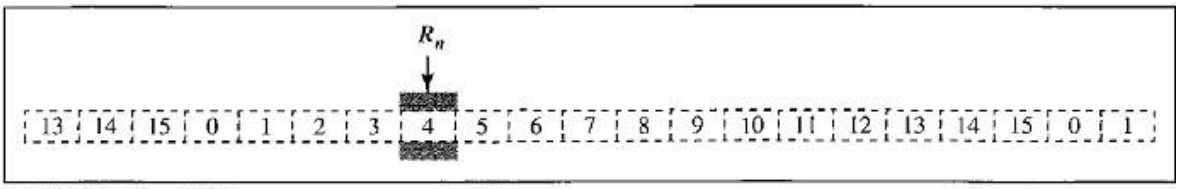
There is only one variable required, that is R_n

R_n: defines next frame expected, belongs to Receive Window.

- The sequence numbers to the left of the window belong to the frames already received and acknowledged.
- The sequence numbers to the right of this window define the frames that cannot be received.
- Any received frame with a sequence number in these two regions is discarded.
- Only a frame with a sequence number matching the value of R_n is accepted and acknowledged. (i.e. $S_n=R_n$)
- The receive window slides only one slot at a time, when the correct frame is received the window slides.



a. Receive window



b. Window after sliding

Timers

The timer for the first outstanding frame always expires first.We send all outstanding frames when this timer expires.

Acknowledgment

- The receiver sends a positive acknowledgment if a frame has arrived safe and sound and in order.
- If a frame is damaged or frame is received out of order, the receiver is silent and will discard all subsequent frames until it receives the one it is expecting.
- The silence of the receiver causes the timer of the unacknowledged frame at the sender site to expire. This causes the sender to go back and resend all frames, beginning with the one with the expired timer.
- The receiver does not have to acknowledge each frame received. It can send one cumulative acknowledgment for several frames.

Retransmission (Resending a Frame)

When the timer expires, the sender resends all outstanding frames.

Example:

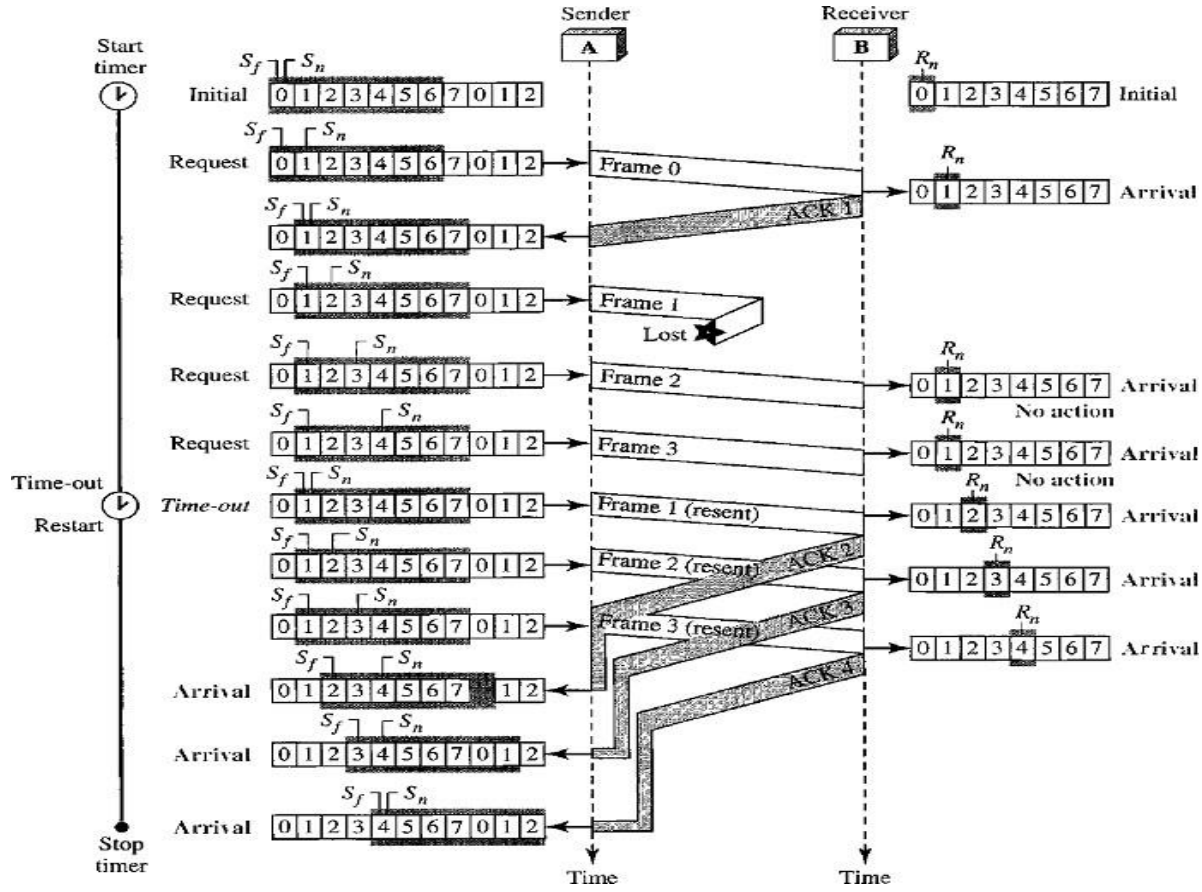
- Suppose the sender has already sent frame 6, but the timer for frame 3 expires.
- This means that frame 3 has not been acknowledged.
- The sender goes back and sends frames 3, 4,5, and 6 again.
- That is why the protocol is called ***Go-Back-N*** **ARQ**.

Flow Diagram

The below shows what happens when a frame is lost.

- Frames 0, 1, 2, and 3 are sent.
- Frame 0 is acknowledged but **Frame 1** is lost.
- The receiver receives frames 2 and 3, but they are discarded because they are received out of order (frame 1 is expected).
- The sender receives no acknowledgment about frames 1, 2, or 3.
- Its timer finally expires. The sender sends all outstanding frames (1, 2, and 3) because it does not know whether the frame is lost or corrupted.
- Note that the resending of frames l, 2, and 3 is the response to one single event.
- When the sender is responding to this event, it cannot accept the triggering of other events.
- This means that when ACK 2 arrives, the sender is still busy with sending frame 3.
- The physical layer must wait until this event is completed and the data link layer goes back to its sleeping state.
- Vertical line in the figure is to indicate the delay.

Note that before the second timer expires, all outstanding frames have been sent and the timer is stopped.



Disadvantage with Go-Back-N ARQ

Go-Back-N ARQ protocol is very inefficient for a noisy link.

- *Go-Back-N* ARQ simplifies the process at the receiver site.
- The receiver keeps track of only one variable, and there is no need to buffer out-of-order frames and they are simply discarded.
- In a noisy link a frame has a higher probability of damage; so it leads to the resending of multiple frames.
- This resending uses more bandwidth and slows down the transmission. This is a major disadvantage.

Solution: When there is just one frame is damaged, then only the damaged frame is resent, instead of resending from **Nth** frame. This will be achieved by using **Selective Repeat ARQ** Protocol.

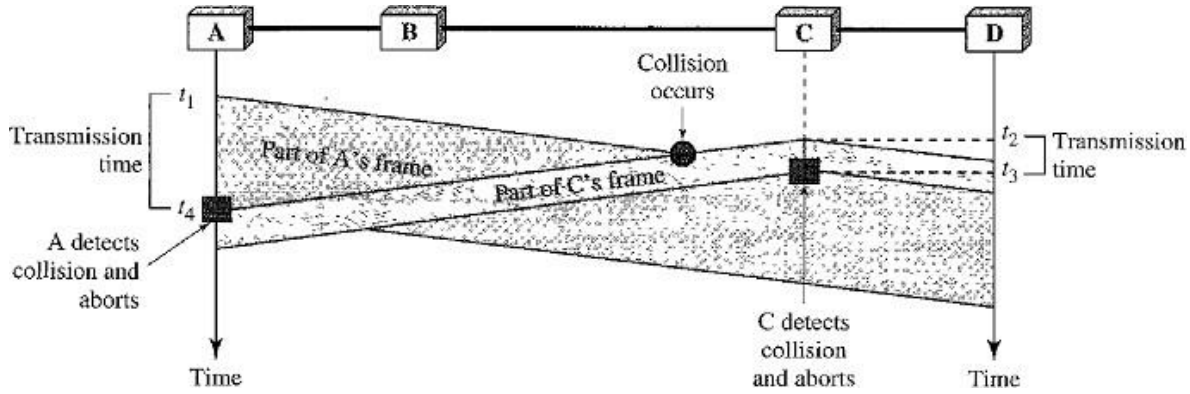
6. What is the purpose of CSMA/CD? And Explain it.

Carrier Sense Multiple Access with Collision Detection (CSMA/CD)

CSMA method does not specify the procedure following a collision but CSMA/CD augments the algorithm to handle the collision.

In this method, a station monitors the medium after it sends a frame to see if the transmission was successful.

- If it is successful the station is finished.
- If it not successful and there is a collision, the frame is sent again.



- At time **t 1**, station A has executed its persistence procedure and starts sending the bits of its frame.
- At time **t2**, station C has not yet sensed the first bit sent by A.
- Station C executes its persistence procedure and starts sending the bits in its frame, which propagate both to the left and to the right.
- The collision occurs sometime after time **t2** Station C detects a collision at time **t3** when it receives the first bit of A's frame. Station C immediately aborts transmission.
- Station A detects collision at time **t4** when it receives the first bit of C's frame; it also immediately aborts transmission.
- A transmits for the duration **t4 – t1**; C transmits for the duration **t3 - t2**
- At time **t4,** the transmission of A‟s frame is aborted; at time t3, the transmission of C's frame is aborted. Both are incomplete.
- This is so because the station, once the entire frame is sent, does not keep a copy of the frame and does not monitor the line for collision detection.

Minimum Frame Size (2Tp)

If the two stations involved in a collision are the maximum distance apart, the signal from the first station takes Tp time to reach the second station and the effect of the collision takes another Tp time to reach the first station.

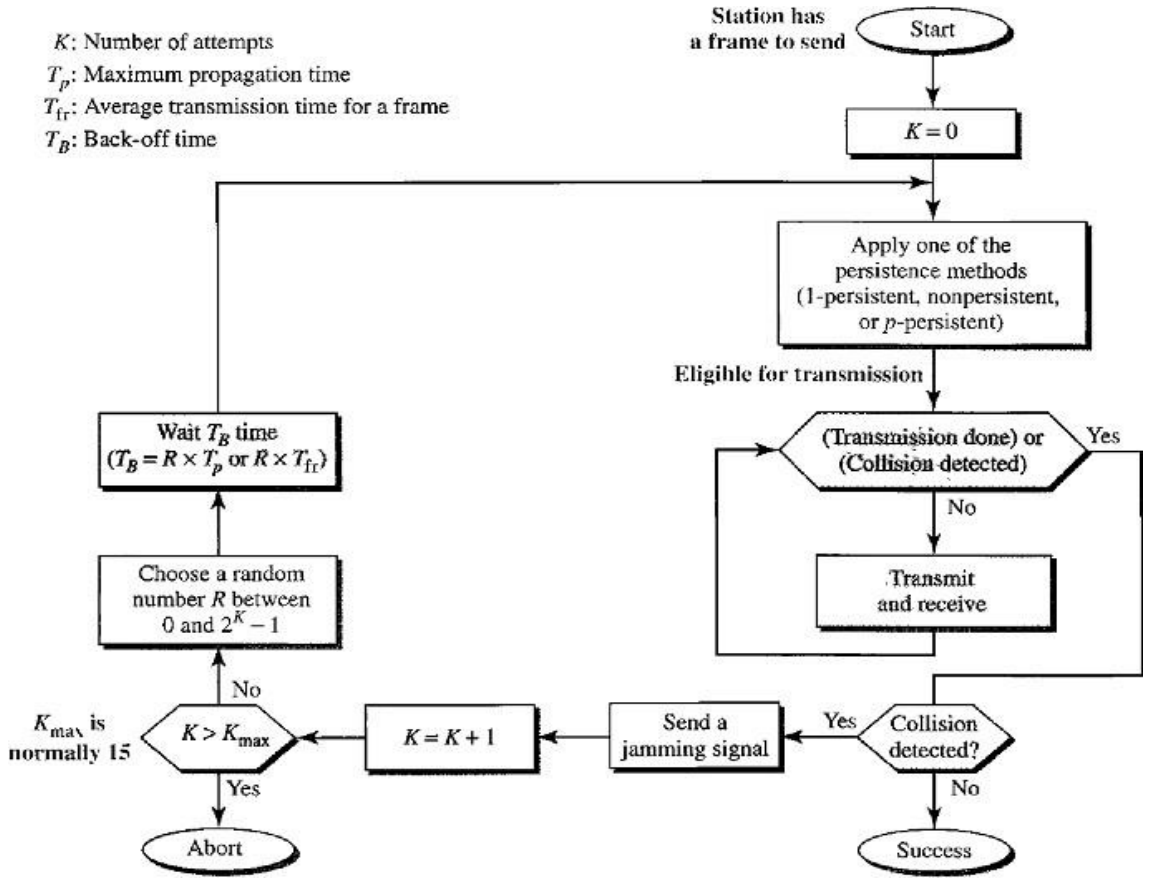
Therefore, the frame transmission time $\mathbf{T_{fr}}$ must be at least two times the maximum propagation time Tp. So the first station must still be transmitting after 2Tp .

Procedure

- We need to sense the channel before we start sending the frame by using one of the persistence processes.
- In ALOHA, we first transmit the entire frame and then wait for an acknowledgment. In *CSMA/CD*, transmission and collision detection is a continuous process.
- We do not send the entire frame and then look for a collision. The station transmits and receives continuously and simultaneously using two different ports.
- We use a loop to show that transmission is a continuous process.
- We constantly monitor in order to detect one of two conditions:

either transmission is finished or a collision is detected. Either event stops transmission.

- When we come out of the loop, if a collision has not been detected, it means that transmission is complete; the entire frame is transmitted. Otherwise, a collision has occurred.
- Here we send a short jamming signal that enforces the collision in case other stations have not yet sensed the collision.

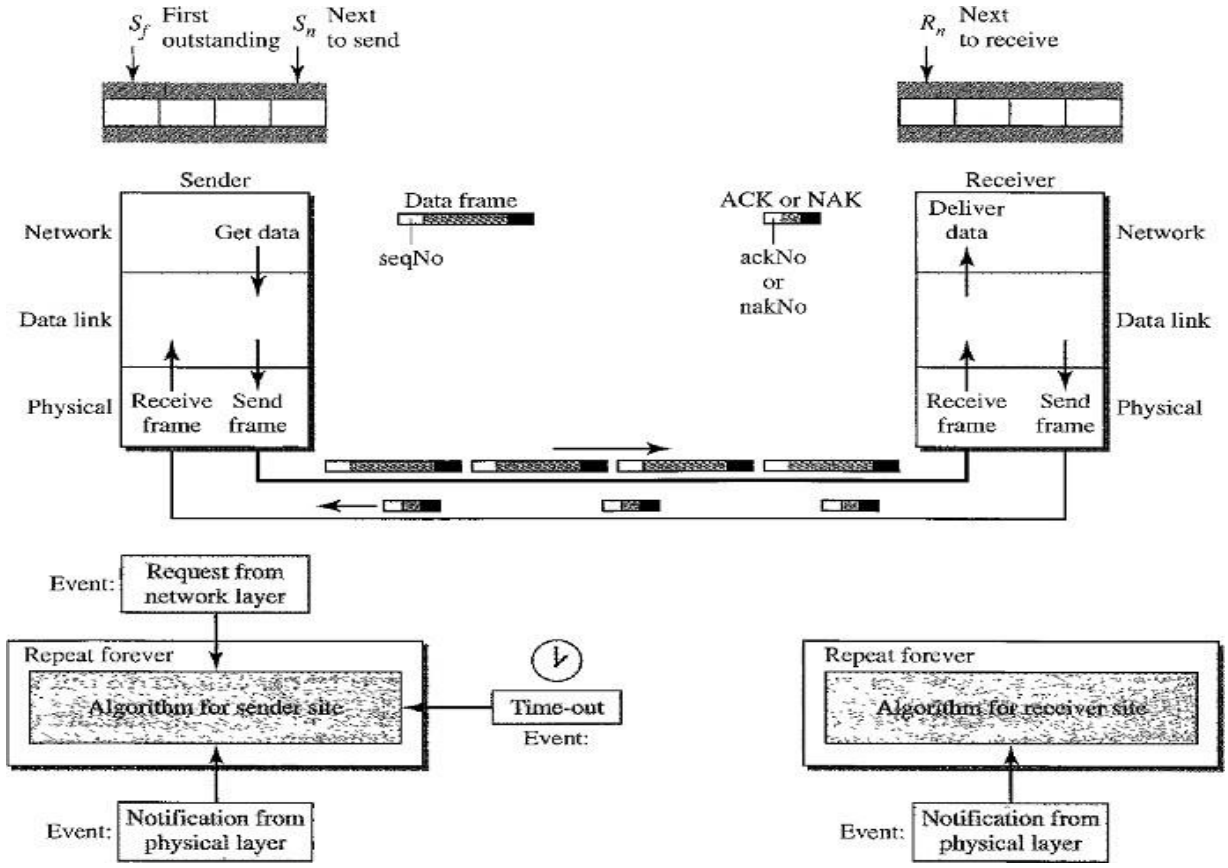


7. Explain and demonstrate Selective repeat sliding window Protocol with an example.

It is more efficient for Noisy links but processing at the receiver is more complex.

Design Window Sizes

Window size 2^{m-1} means the size of the sender and receiver window is at most one half of 2^m **(i.e** 2^m/2).



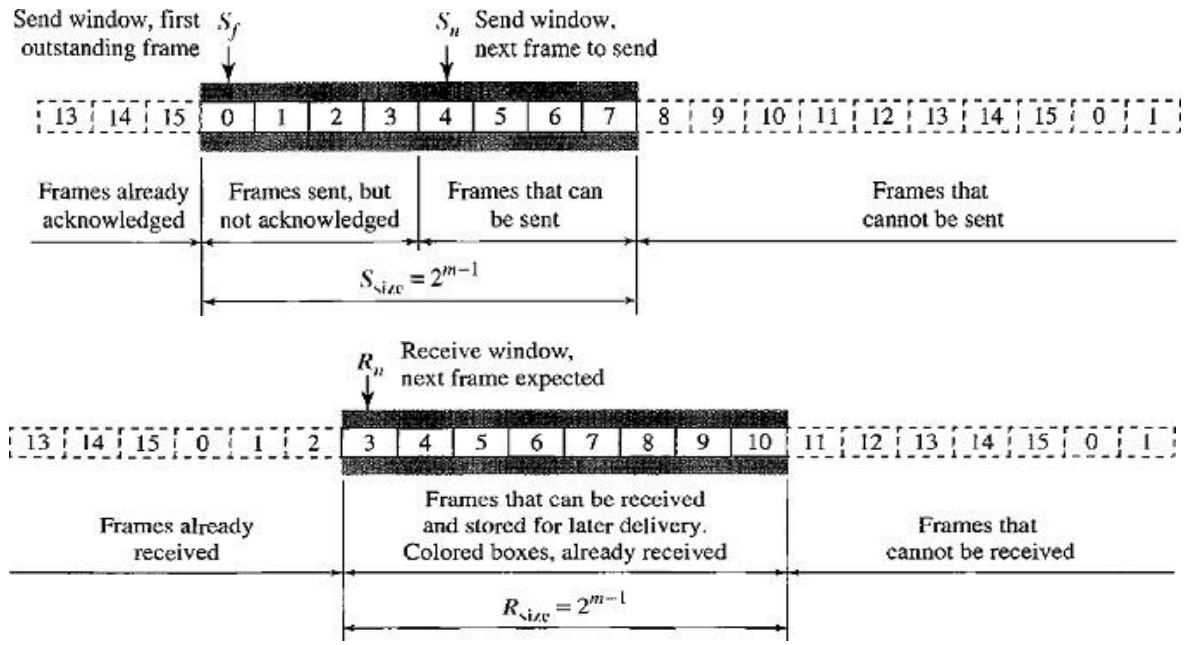
Windows

It uses two windows:

1. Send Window
2. Receive Window

The size of the send window and receive window is same as 2^{m-1}.

Example: If $m = 4$, the sequence numbers ranges from 0 to 15, but the size of the window is just 8.



- The Selective Repeat Protocol allows as many frames as the size of the receive window to arrive out of order and be kept until there is a set of in-order frames to be delivered to the network layer.
- Because the sizes of the send window and receive window are the same, all the frames in the send frame can arrive out of order and be stored until they can be delivered.
- We need to mention that the receiver never delivers packets out of order to the network layer.
- Those slots inside the receive window that are colored define frames that have arrived out of order and are waiting for their neighbors to arrive before delivery to the network layer.

Sender site:

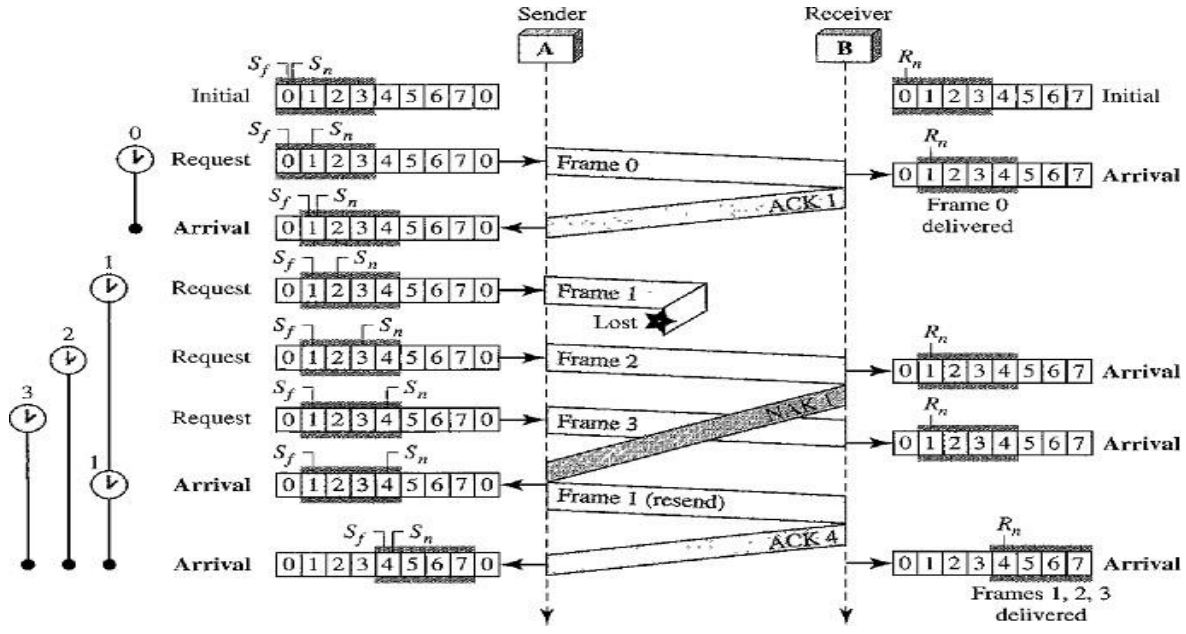
- For every request event a timer is started before sending any frame. The arrival event is more complicated. An ACK or a NAK frame may arrive at the sender site.
- If a valid NAK frame arrives, we just resend the corresponding frame.
- If a valid ACK arrives 3 actions will be done:
  1. Purge the buffers
  2. Stop the corresponding timer
  3. Move the left wall of the window.
- The time-out event is simpler, only the frame which times out is resent.

Receiver site:

- Receiver sends ACK and NAK frames to sender.
- If the Receiver receives a corrupted frame and a NAK has not yet been sent, Receiver sends a NAK to tell the sender that we have not received the frame we expected.
- If the frame is not corrupted and the sequence number is in the window, Receiver stores the frame and marks the slot.
- If contiguous frames, starting from R_n have been marked, Data link layer delivers their data to the network layer and slide the window.

Flow Diagram

By looking at the below flow diagram, we can observe below points:



Timers

- Each frame sent or resent needs a timer and the timers needs to be numbered such as 0,1,2,3..etc.
- The timer for frame 0 starts at the first request, but stops when the ACK for this frame arrives.
- The timer for frame 1 starts at the second request and restarts when a NAK arrives, and finally stops when the last ACK arrives.
- The other two timers start when the corresponding frames are sent and stop at the last arrival event.

Receiver Site

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

Importance of NAK's

- Here a NAK is sent after the second arrival, but not after the third.
- The reason is that the protocol does not want to crowd the network with unnecessary NAKs and unnecessary resent frames.
- The second NAK would still be NAK1 to inform the sender to resend frame 1 again; this has already been done.
- The first NAK sent is remembered and is not sent again until the frame slides.
- A NAK is sent once for each window position and defines the first slot in the window.

Importance of ACK's

- There are only two ACKs are sent here. The first one acknowledges only the first frame The second one acknowledges three frames.
- In Selective Repeat, ACKs are sent when data are delivered to the network layer.
- If the data belonging to *n* frames are delivered in one shot, only one ACK is sent for all of them.
- In the above figure frame 1,2,3, are sent to Network layer and then ACK4 is sent, to represent that Frame1, Frame2, Frame3 are delivered.

There is a question arises that :

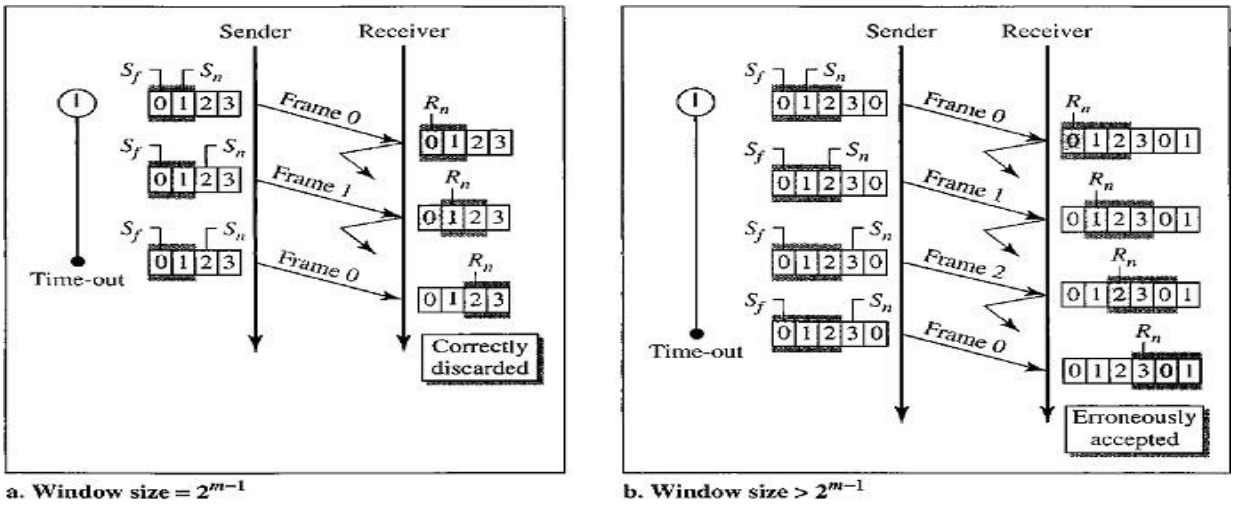
Why the size of the sender and receiver windows is 2^{m-1}?

Sol: For an example, take m = 2, which means the size of the window is **$2^m/2$**, or 2.

Now we compare window size=2 and window size =3.

Window size=2

- If the size of the window is 2 and all acknowledgments are lost, the timer for frame 0 expires and frame 0 is resent.
- The window of the receiver is now expecting frame 2, not frame 0, so this duplicate frame is correctly discarded.



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

Window Size=3

- When the size of the window is 3 and all acknowledgments are lost, the sender sends a duplicate of frame 0.
- However, this time, the window of the receiver expects to receive frame 0 (0 is part of the window), so it accepts frame 0, not as a duplicate, but as the first frame in the next cycle.
- This is clearly an error.

8. Explain the functions of various connecting devices:

1. Repeater – A repeater operates at the physical layer. Its job is to regenerate the signal over the same network before the signal becomes too weak or corrupted so as to extend the length to which the signal can be transmitted over the same network.

2. Hub – A hub is basically a multiport repeater. A hub connects multiple wires coming from different branches, for example, the connector in star topology which connects different stations. Hubs cannot filter data, so data packets are sent to all connected devices.

3. Bridge – A bridge operates at data link layer. A bridge is a repeater, with add on the functionality of filtering content by reading the MAC addresses of source and destination. It is also used for interconnecting two LANs working on the same protocol. It has a single input and single output port, thus making it a 2 port device.

- **Transparent Bridges:-** These are the bridge in which the stations are completely unaware of the bridge's existence i.e. whether or not a bridge is added or deleted from the network, reconfiguration of the stations is unnecessary. These bridges make use of two processes i.e. bridge forwarding and bridge learning.

4. Switch – A switch is a multiport bridge with a buffer and a design that can boost its efficiency(a large number of ports imply less traffic) and performance. A switch is a data link layer device. The switch can perform error checking before forwarding data, that makes it very efficient as it does not forward packets that have errors and forward good packets selectively to correct port only.

5. A **router** is a **network layer** hardware device that transmits data from one LAN to another if both networks support the same set of protocols. So a **router** is typically connected to at least two LANs and the **internet service provider** (ISP). It receives its data in the form of **packets**, which are **data frames** with their **destination address** added. Router also strengthens the signals before transmitting them. That is why it is also called **repeater**.

Routing Table

A router reads its routing table to decide the best available route the packet can take to reach its destination quickly and accurately. The routing table may be of these two types −

- **Static** − In a static routing table the routes are fed manually. So it is suitable only for very small networks that have maximum two to three routers.
- **Dynamic** − In a dynamic routing table, the router communicates with other routers through protocols to determine which routes are free. This is suited for larger networks where manual feeding may not be feasible due to large number of routers.

6. Gateway

Gateway is a network device used to connect two or more dissimilar networks. In networking parlance, networks that use different protocols are **dissimilar networks**. A gateway usually is a computer with multiple **NICs** connected to different networks. A gateway can also be configured completely using software. As networks connect to a different network through gateways, these gateways are usually hosts or end points of the network.

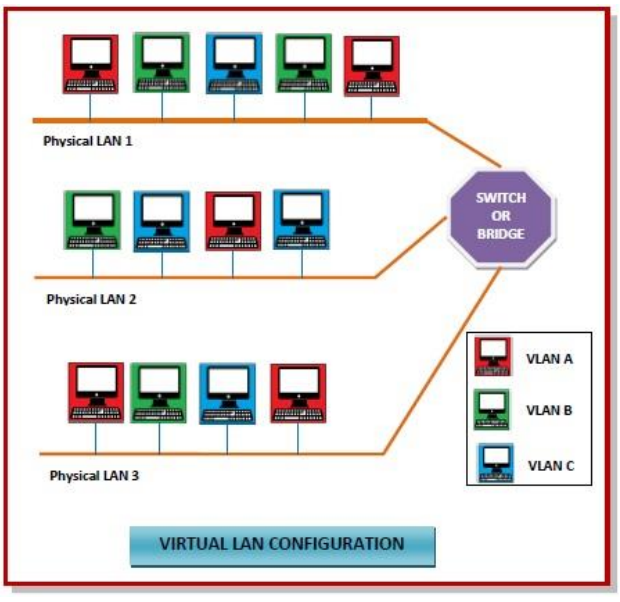
Gateway uses **packet switching** technique to transmit data from one network to another. In this way it is similar to a **router**, the only difference being router can transmit data only over networks that use same protocols.

VLANS

Virtual Local Area Networks or Virtual LANs (VLANs) are a logical group of computers that appear to be on the same LAN irrespective of the configuration of the underlying physical network.

Network administrators partition the networks to match the functional requirements of the VLANs so that each VLAN comprise of a subset of ports on a single or multiple switches or bridges.

This allows computers and devices in a VLAN to communicate in the simulated environment as if it is a separate LAN.



- A VLAN forms sub-network grouping together devices on separate physical LANs.
- VLAN's help the network manager to segment LANs logically into different broadcast domains.
- VLANs function at layer 2, i.e. Data Link Layer of the OSI model.
- There may be one or more network bridges or switches to form multiple, independent VLANs.
- Using VLANs, network administrators can easily partition a single switched network into multiple networks depending upon the functional and security requirements of their systems.
- VLANs eliminate the requirement to run new cables or reconfiguring physical connections in the present network infrastructure.