4. In this process one way is to add the sender address and destination MAC address
5. The sender address will help to acknowledge the source and the destination address to help frame where it must reach.
6. If frame size is very large then any single bit error we have to go for the retransmission of the complete frame and also makes the flow control and error control inefficient.
7. At receiver check sum is once again calculated and compared with the one at the source.
8. If they are different the data link layer will discard and ask for retransmission of the frame.

3.3.1 Character count

- The frame size maybe fixed or variable.
- Here the size itself is used as the delimiters. Ex: ATM, WAN.
- Variable size framing is used in local area network.
- This method uses a header field to specify the number of characters in the frame.
- When data link layer at the destination checks the character count it knows how many characters follow and hence where the end of the frame.

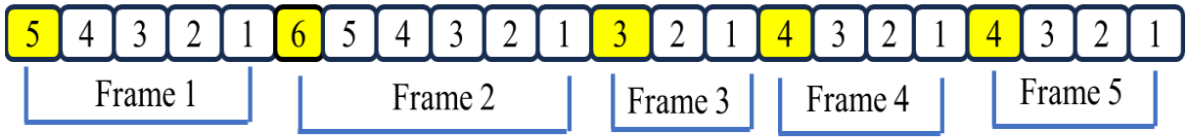


Fig.3.13. Character count method of framing.

- The trouble in this is that the count can be garbled by a transmission error.
- Let us assume that in the precious example the second frame in having 6 bits and some error occurred and it is changed to 5. The problem of it is as shown below.

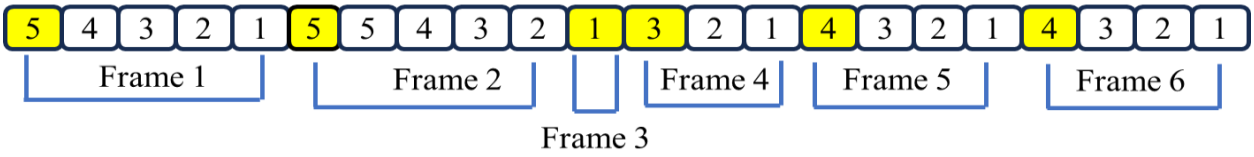


Fig.3.14. Transmission error occurs in Character count method of framing.

- In In the above case, we can understand that in the frame 2 the header count has been changed from 6 to 5 due to transmission error. So, the receiver has counted five data bits for frame 2.
- Because of which the size of the frame 3 has been reduced to one bit and the receiver has counted as frame 3 is having only one bit.
- Here sending a frame back and asking for retransmission does not help here as destination is not known and we don't know how many characteristics to skip, to start retransmission. So, it is used rarely.

3.3.2 Flag bytes with byte stuffing

- In this method data is framed as 8-bit characteristics (using ASCII coding).

- Each frame is having a header which is giving source and destination address.
- The tailer carries the other control inform which composes of error detection bits. There are also multiple of 8 bits.
- Here to separate from one frame to another an 8-bit flag is added at the beginning and end of the frame.

Fig.3.15. Flag bytes with byte stuffing.

- Character oriented framing was popular when only text was exchanged by the data link layer. The flag could be selected to be any character which is not used for text communication.
- We also send other types of information such as graphs, audio and video and any character used for the flag could also be part of the information.
- In such scenario the receiver will come to a decision that it has reached the end of the frame in the middle of the data.
- In such cases byte-stuffing strategy was added to character-oriented framing.
- In byte stuffing, a special byte is added to the data session of the frame when there is a character with the same pattern as the flag. The data session is added as an extra byte.
- This byte is usually called as escape character 'ECS' and has a predefined bit pattern.
- Whenever receiver encounters the ESC, it removes from the data section and treat the next character as data, not a delimiting flag.

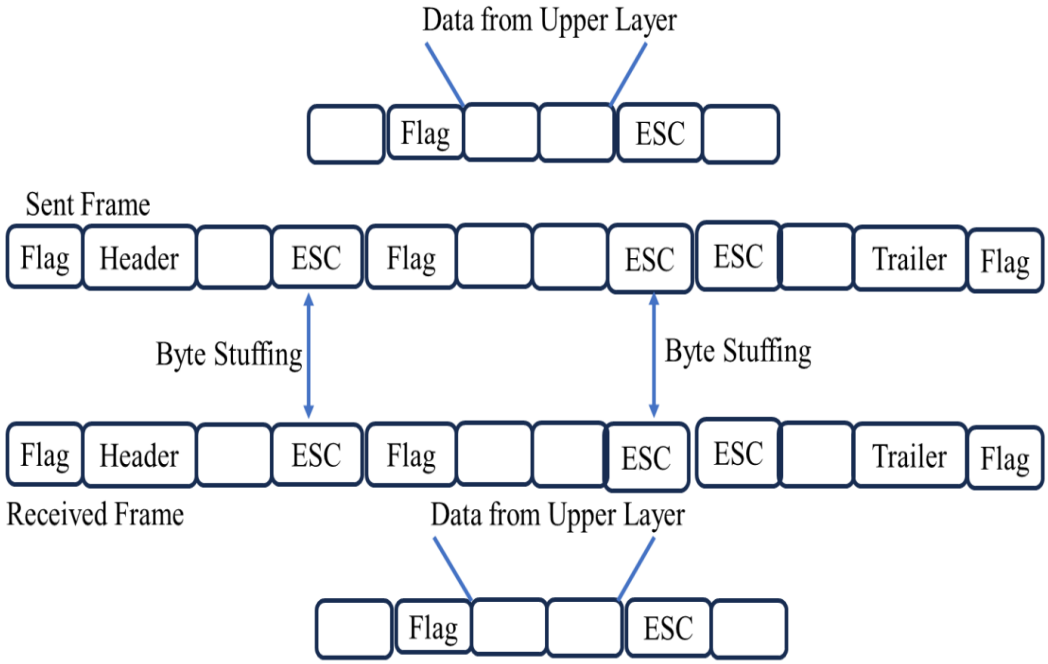


Fig.3.16. Example of Flag bytes with byte stuffing.

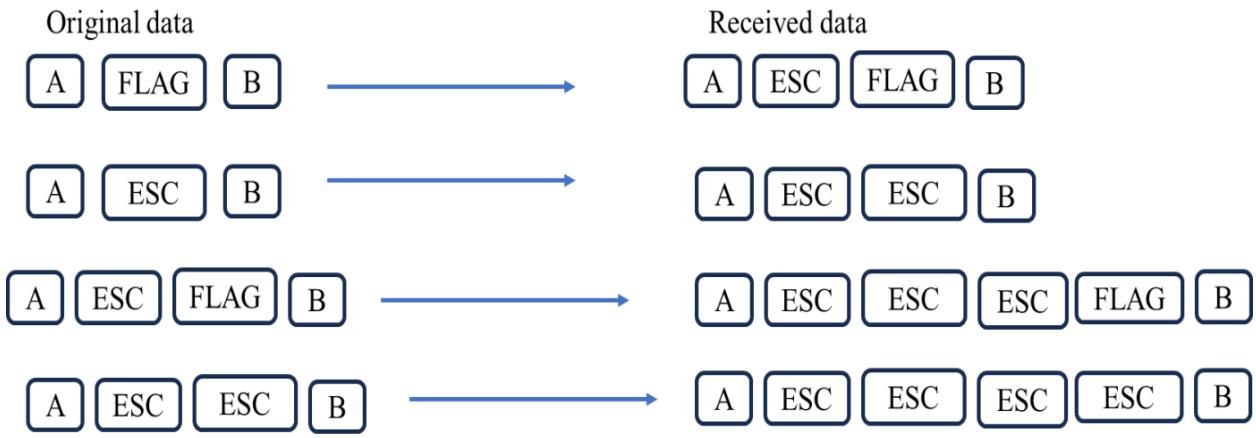


Fig.3.17. Another example of Flag bytes with byte stuffing.

3.3.3 Starting and Ending Flags with bit stuffing.

1. Character oriented protocols present another problem in data communication.
2. The universal coding system which we use today such as Unicode have 16 bit and 32 bit characteristics that conflict with 8 bit characters.
3. So, in such scenario the data section of a frame is a sequence of bits from upper layer as text graphics audio video and so on. However, in addition to the headers we will find the flag fields in the payload or data.
4. Most protocols use special 8-bit pattern called as flag, 0111110 delimiter to define the beginning and end of the frame.
5. This flag can create the same type of pattern as that seen in a byte stuffing.
6. To distinguish between data and flag we add zero after five consecutive ones in the data.
7. The extra stuffed bit is removed from the data by the receiver. This method is called bit stuffing and here a 0 is added if a zero is followed by 5 ones.
8. The real flag is not stuffed with zero by the sender to distinguish data and flag.

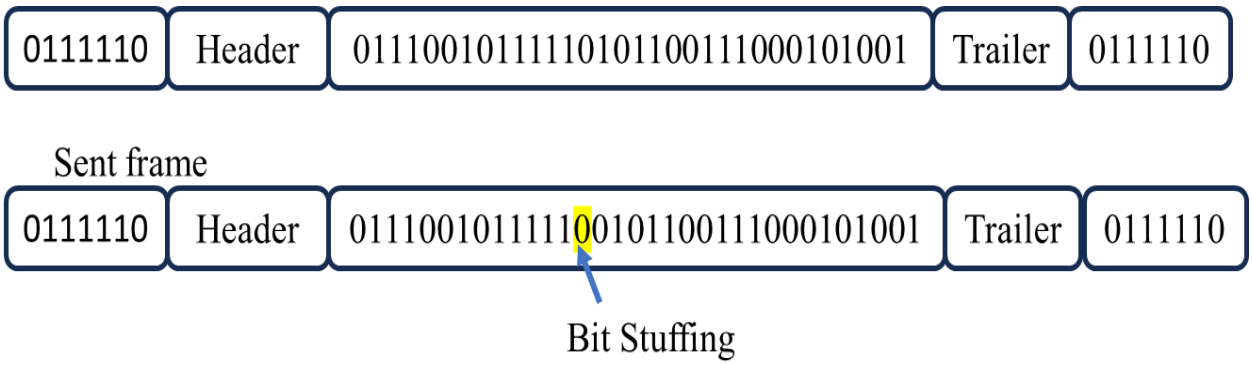


Fig.3.18. Example of Flag bytes with bit stuffing.

3.4 Physical layer violations

1. This framing method is used only in those networks in which encoding on the physical medium contain some redundancy.
2. Some LANs encode each bit of data by using two physical bits loke Manchester coding uses. Here, Bit 1 is encoded into a high-low (10) pair and Bit 0 is encoded into a low-high (01) pair.
3. The scheme means that every data bit has a transition in the middle, making it easy for the receiver to locate the bit boundaries.
4. The combinations high-high and low-low are not used for data but are used for delimiting frames in some protocols.

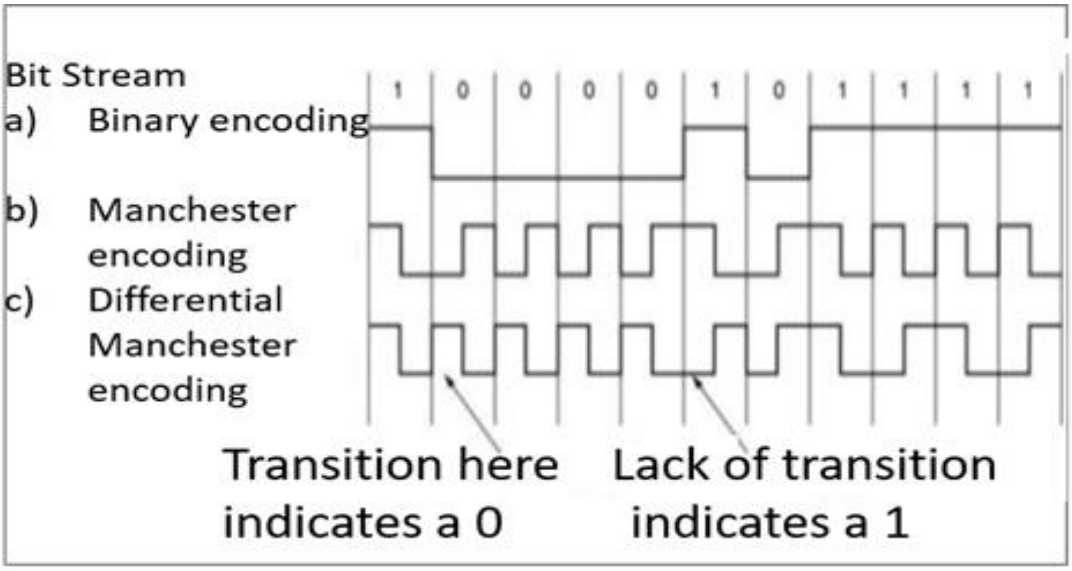


Fig.3.19. Example of Physical layer coding violations.

3.4 Flow Control

1. Flow control – controls the amount of data sent before receiving acknowledgment.
2. This is the most important duty of the link layer.
3. It also sets the procedures that tell a sender how much data it can transmit, before it must wait for an acknowledgment from the receiver.
4. Incoming data is to be processed and checked before it can be used. The rate of such processing is always slower than the rate of transmission. So, receiving devices have limited processing speeds.
5. Receiving devices have a block of memory called buffer in which they store the incoming data until they are processed.
6. The receiving device must inform the sender regarding the processing and memory limits and should inform the sender to slow down or stop, respectively.

3.4.1 Stop and Wait protocol.

1. Stop and wait is the simplest flow control mechanism.
2. The sending device must have copy of the last frame, this allows for retransmission in cases of lost and damaged frames until they are received correctly.
3. For identification both the transmitted and acknowledged frames are numbered automatically.
4. "0" frame is acknowledged by an Ack "1" frame indicating that it is expecting data frame 1.
5. This numbering helps in case of duplicate transmission.
6. A damaged or lost frame is treated in the same manner by the receiver.
7. If the receiver detects an error, it discards the frame and sends no acknowledgement.
8. If the receiver receives an out of order frame it discards the out of order received frame.
9. The sender has a control variable S that holds the number of recently sent frame (0 or 1). Similarly, the receiver had a control variable called R that holds the number of the next frame (0 or 1).
10. When the frame is sent the sender starts a timer. If the acknowledgement is not received in the given time, the sender assumes that the frame is lost or damaged.
11. At the sender the device will be having a memory named S in which the transmitted frames will be stored and at the receiver the received frames will be stored a memory named R.
12. During transmission the sender and receiver will update the frame numbers to the numbers of the frames to be transmitted and received.

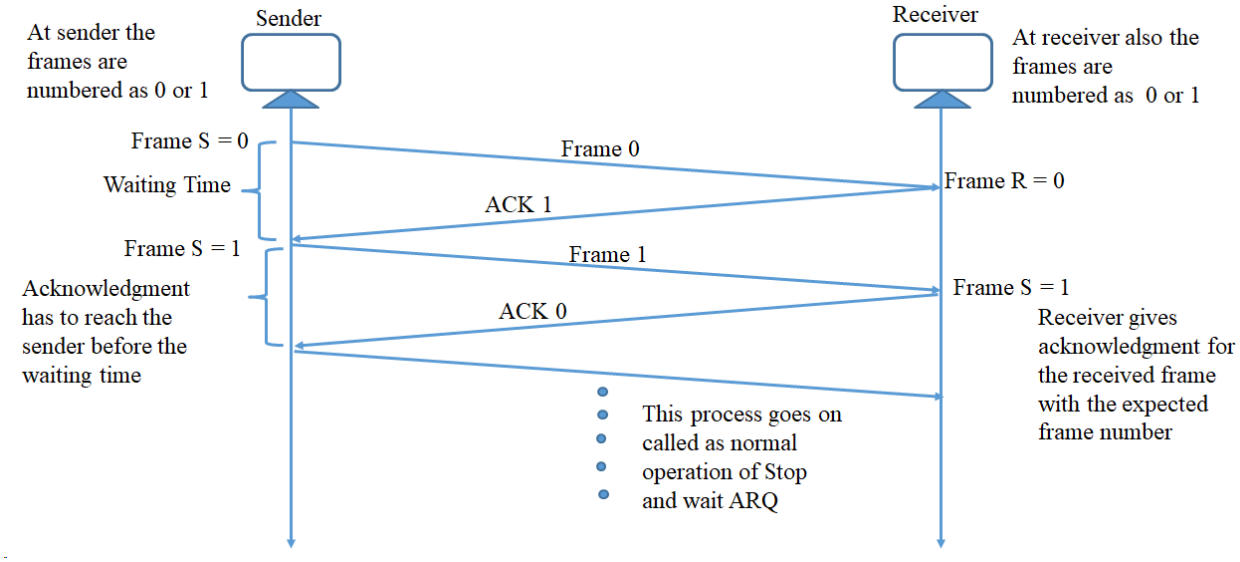


Fig.3.20. Normal operation of Stop and wait protocol.

Operation

When frames are transmitted, we have 4 different cases due to transmission errors. They are 1. Normal Operation 2. Lost or damaged frame 3. Lost acknowledgment 4. Delayed acknowledgment.

1. Normal Operation

1. In normal operation the sender sends frame ‘0’ and waits for acknowledgement ‘1’.
2. The transmitter sets a timer and waits for the acknowledgement for frame ‘0’ i.e., ACK ‘1’
3. When ACK ‘1’ one is received it sends frame ‘1’ and then waits to receive ACK ‘0’ and so on. The acknowledgement must be received before the timer expires for each frame.
4. The normal operation of stop and wait protocol is given in figure 3.20.

2. Lost or damaged frame

1. A lost or damaged frame is handled in the same way by the receiver.
2. When the receiver receives a damaged frame, it discards it, potentially means that frame is lost during transmission.
3. The receiver remains silent about the last frame and keeps its value of R unchanged.

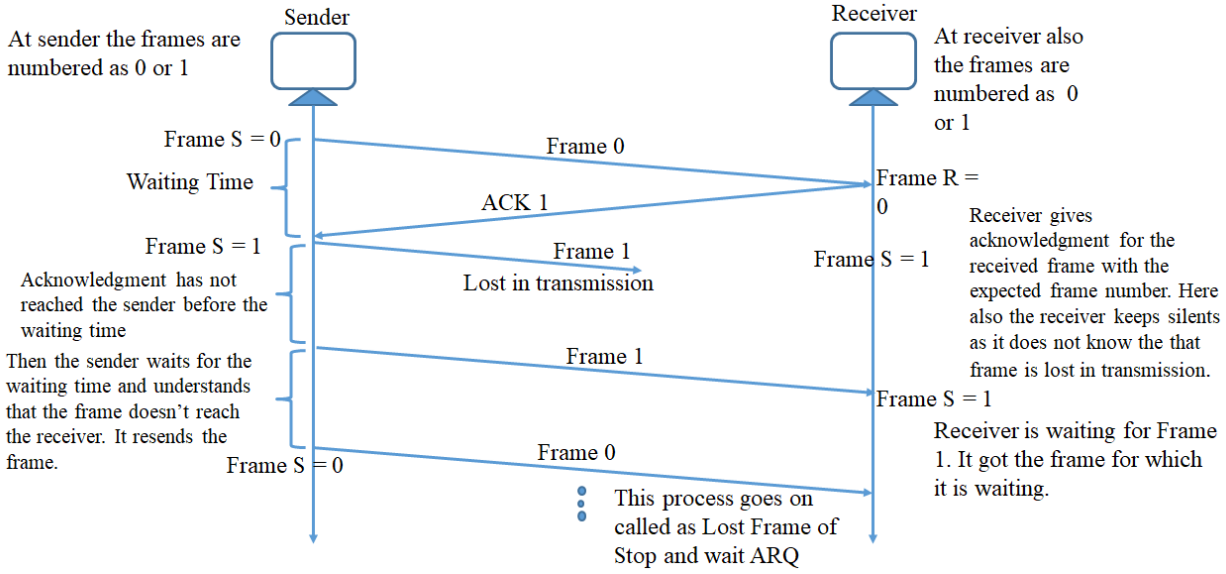


Fig.3.21. Lost or damaged frame operation of Stop and wait protocol.

3. Lost acknowledgement

1. A lost or damaged acknowledgement is handled in the same way by the sender.
2. When the acknowledgment is lost then the sender will wait until the timer is completed and then it retransmits the same frame.
3. If the sender receives a damaged acknowledgement, it discards it.
4. When the timer for frame ‘1’ expires the sender retransmits the frame ‘1’.
5. Note that the receiver has already received the frame ‘1’ and expecting to receive frame ‘0’ so it silently discards the second copy of frame ‘1’.

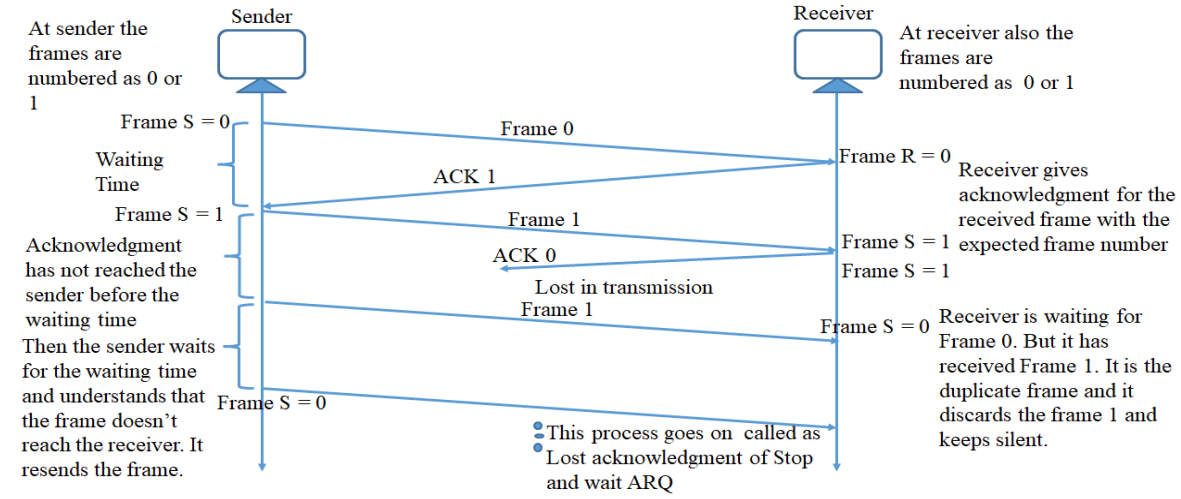


Fig.3.22. Lost acknowledgment operation of stop and wait protocol.

4. Delayed acknowledgement

1. An acknowledge can delayed at the receiver or by some problem in the link.
2. If the sender receives The acknowledgement after the time is out then it retransmits the frame.
3. Let us suppose the sender gets the acknowledgement after retransmitting the frame one.
4. This frame one is lost in transmission and the Timer for frame one switched again and waits for timer to end.
5. The receiver will not send a new acknowledgement as it has already sent an acknowledgement and keep silent.
6. If a duplicate copy of frame one is received then the receiver discards the frame.

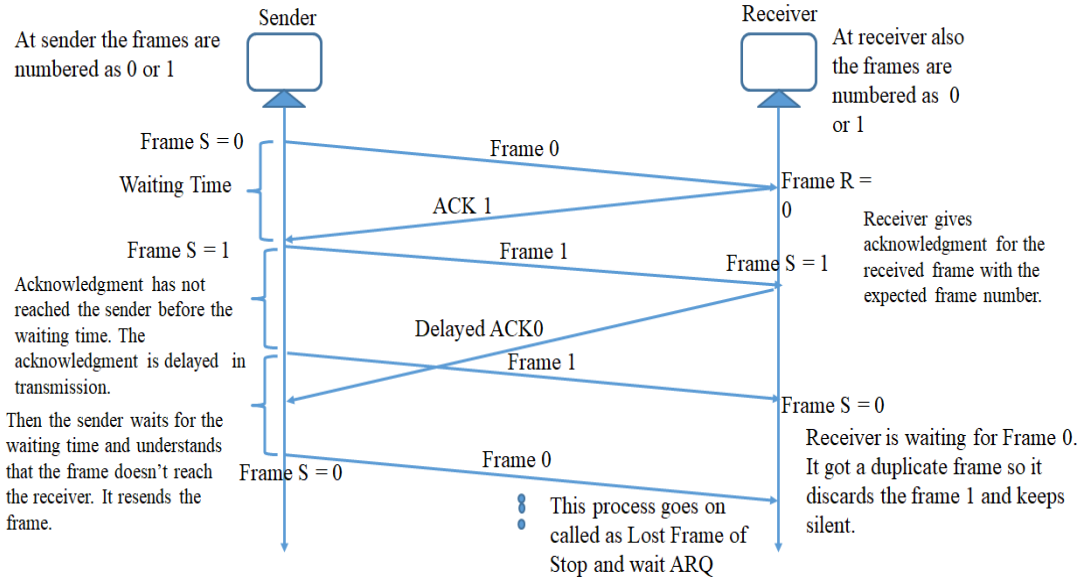
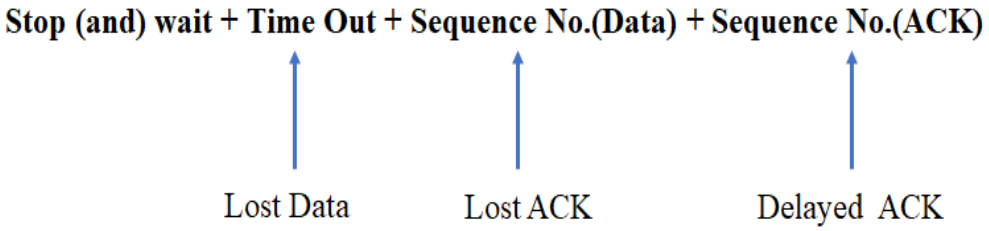


Fig.3.23. Delayed acknowledgment operation of stop and wait protocol.

The above 3 problems are resolved by Stop and Wait. So, only this protocol has another name

called Automatic repeat Request. This protocol does both error control and flow control.



3.4.2 Go-Back-N ARQ

1. In Stop and wait protocol before a sending a frame the sender is waiting for acknowledgment.
2. That means link is unused till acknowledgment is received. This is not a proper use of the transmission link. So sending of multiple frames is a good use to improve the efficiency of the transmission link.
3. In Go-Back-N ARQ we send multiple frames before getting a acknowledgment.
4. Frames are sequentially numbered at the sender. So we should have limit for the sequence number. This depends on the number of bits allowed for the sequence number.
5. If the number of bits allocated in the frame header are 'm' then the range of the sequence bits is 0 to 2^m-1.
6. These sequence numbers are repeated. The frames are stored in the buffer at sender side.
7. A sliding window is used to update the status of the buffer when frames are sent.
8. The start and stop of the window are represented by S_F, S_L and S represents the frame being transmitted. An example of the sliding window is given below. S_F, S_L and S are called control variables.
9. In the given example m = 3 so the range of sequence numbers is 0-7.
10. In the figure given below the pink colour of the frame represents that the frames are sent, and acknowledgment is received. These frames can be removed from the buffer.
11. Yellow colour of the frame represents the frames are still to be transmitted.
12. The frames which are not coloured are being transmitted at the sender side.
13. At the receiver the size of the sliding window is one.

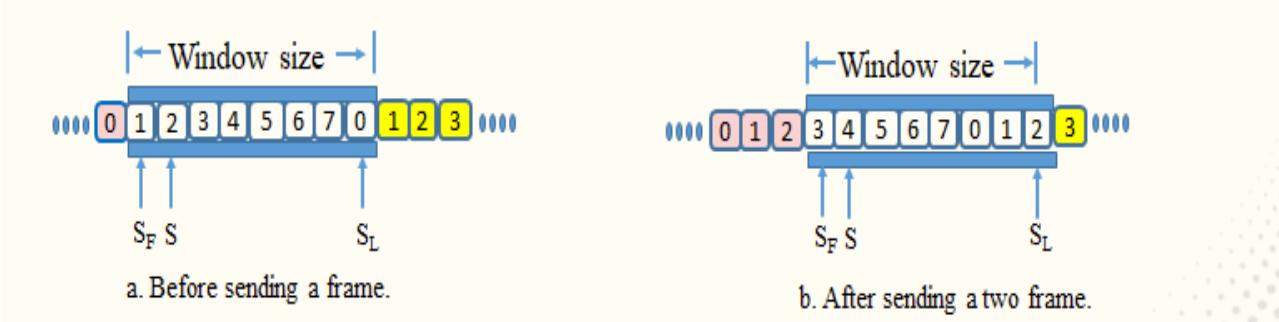


Fig.3.24. Sender sliding window before and after sending frames.

14. The receivers checks for a specific frame to receive in order.
15. The frames received are out of order are discarded at the receiver.
16. Here the pink frames represent that the frame are already acknowledged.
17. Yellow colour frames are yet to be received and acknowledged.
18. The frames which are not coloured are to be received by the receiver.

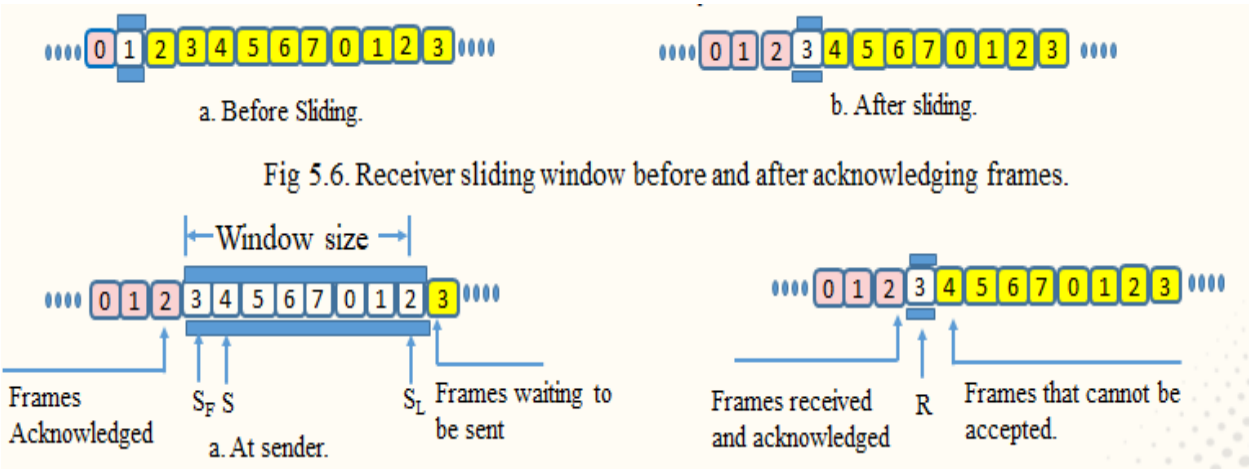


Fig.3.25. Control variables for Go-Back-N ARQ.

Operation

When frames are transmitted, we have 4 different cases due to transmission errors. They are 1. Normal Operation 2. Lost or damaged frame 3. Lost acknowledgment 4. Cumulative acknowledgment 5. Delayed acknowledgment.

1. Normal Operation

1. In normal operation the sender sends frame '0' and waits for acknowledgement '1'.
2. The transmitter sets a timer and waits for the acknowledgement for frame '0' i.e., ACK '1'
3. When ACK '1' one is received it sends frame '1' and then waits to receive ACK '0' and so on. The acknowledgement must be received before the timer expires for each frame.
4. Normal operation in Go-Back-N is given in Fig.3.26.

2. Lost or damaged frame

1. A lost or damaged frame is handled in the same way by the receiver.
2. When the receiver receives a damaged frame, it discards it, potentially means that frame is lost during transmission.
3. The receiver remains silent about the last frame and keeps its value of R unchanged.
4. In Go-Back-N protocol, when the sender receives a ACK for a frame which has been send, then it understands that the frame is lost in transmission or damaged.

5. It them, goes back to that frame and retransmits all the frames from that frame.
6. In the example given below the receiver gives ACK 2 even though the frame is transmitted.
7. Now once the sender receives the ACK 2 it goes back to frame 2 and retransmits the remaining all frames from there. So it is called Go-Back-N protocol. The Fig.3.27

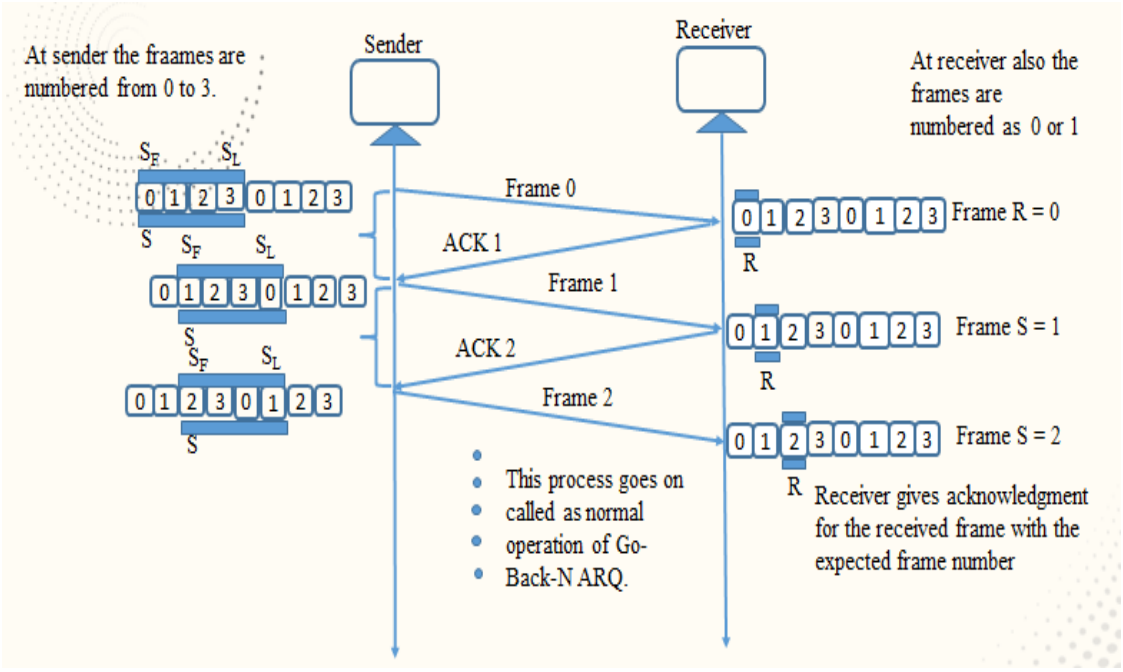


Fig.3.26. Normal operation of Go-Back-N protocol.

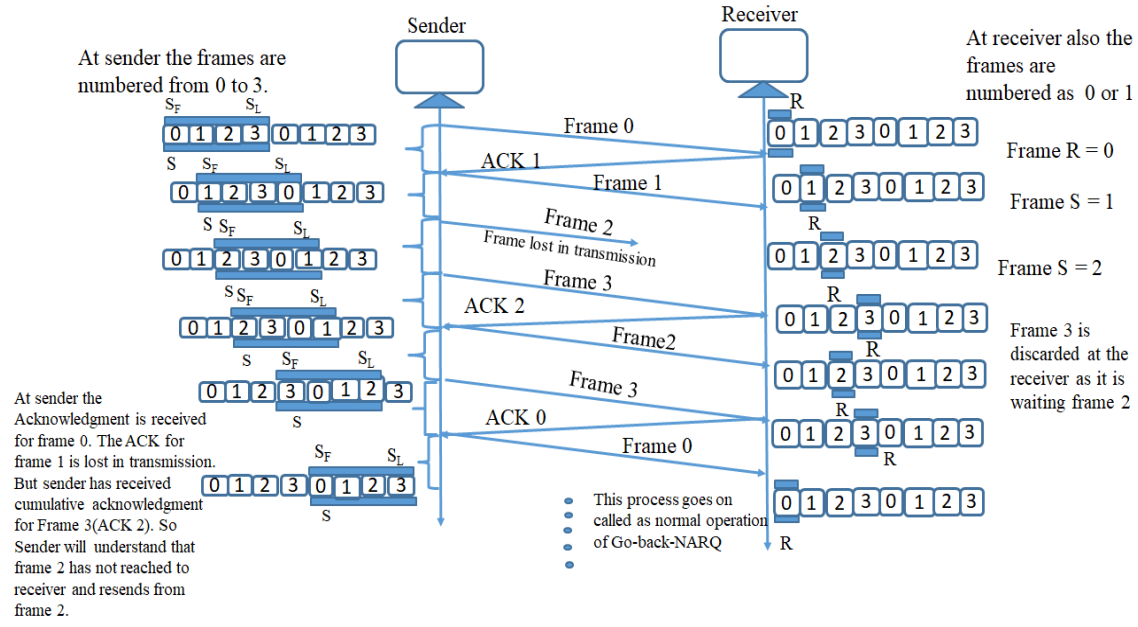


Fig.3.27. Lost and damaged frame operation of Go-Back-N protocol.

3. Lost acknowledgment

1. Lost or damaged acknowledgement is handled in the same way by the sender.
2. When the acknowledgment is lost then the sender will wait until the timer is completed and then it retransmits the same frame.
3. In Go-Back-N the sender will go back to the last acknowledgement received and retransmit all the frames from that frame.
4. If the sender receives a damaged acknowledgement, it discards it and does the same as it has done for lost acknowledgement.
5. In the example given below ACK 2 is lost in transmission. The sender has received only ACK 1 so the sender went back to frame 1 and retransmitted all the frames from there i.e., from frame 1 to frame 3.
7. The receiver will remain silent in this case because it has given an acknowledgement already.
8. The receiver will discard all the duplicate frames and sends ACK 0 as it is waiting for the next set of frames to be transmitted.

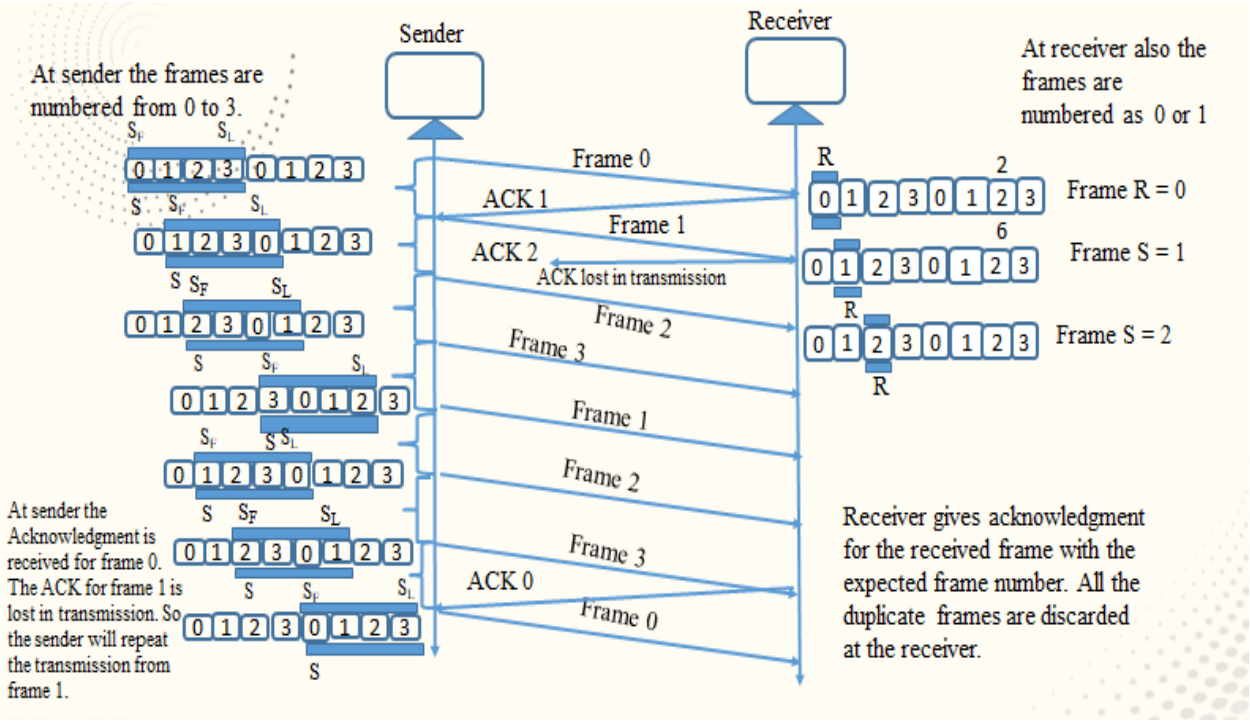


Fig.3.28. Lost and damaged acknowledgment operation of Go-Back-N protocol.

4. Cumulative acknowledgment

1. Cumulative acknowledgement means the receiver sends an acknowledgment after it has received set of frames.
2. Let us suppose that the receiver has received N number of names then it will send a community acknowledgement with N+1 stating that all the N frames has been received by

the receiver are without any error.

3. Such type of cumulative acknowledgement is used in Go-Back-N protocol.
4. Then sender will understand that it has to send the frames from N+1.
5. The cumulative acknowledgement is implemented by the receiver the considering a timer at the receiver.
6. Once the receiver receives a frame it starts the timer, and this timer is smaller than the timer at the sender.
7. After the timer expires the receiver sends a cumulative acknowledgement of all the frames that are unacknowledged till that moment.
8. Go back and uses independent acknowledgements and cumulative acknowledgments depending upon the expiry of the timer.
9. The flow diagram for the cumulative acknowledgement case.

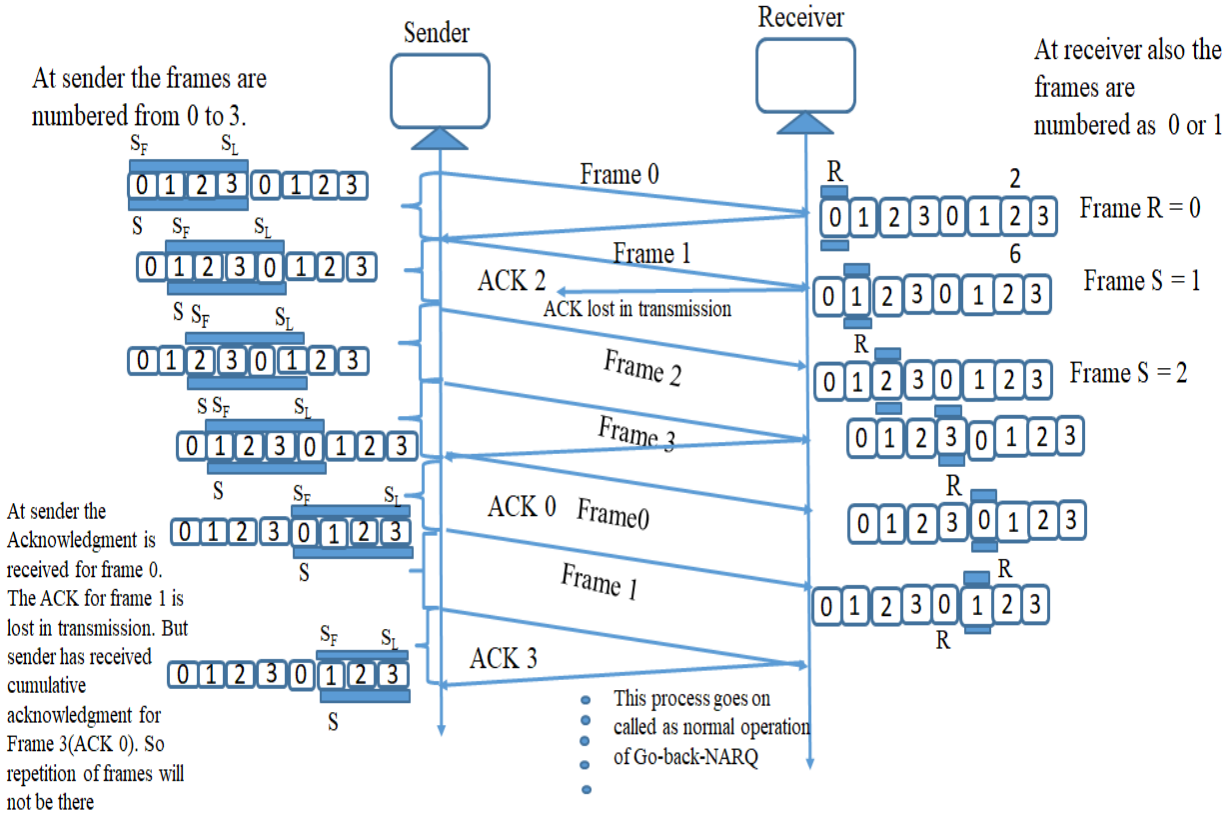


Fig.3.29. Cumulative acknowledgment operation of Go-Back-N protocol.

5. Delayed acknowledgment

1. As Go-Back-N protocol uses both cumulative and individual acknowledgements, delayed acknowledgement may not become a bigger problem.
2. If any case, acknowledgement is delayed in case of individual acknowledgements the receiver will discard all the new frames and will not update the receiver window.
3. In such case the sender will Go-Back to the last acknowledgement received and retransmits all the frames.

3.4.3 Selective repeat request

1. In selective repeat ARQ the sender sliding window is same as that of the sender sliding window in Go-Back-N.
2. The receiver window now will be having the control variables R_F and R_L representing the first and last frame in the window. "R" will be representing the frame to be received.
3. The control variables for Selective repeat ARQ are given below.

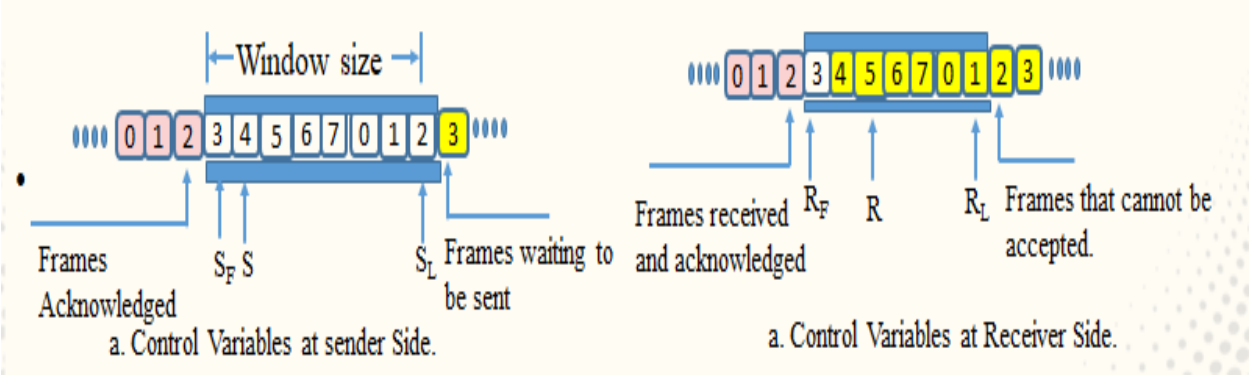


Fig.3.30. Control variables for selective repeat ARQ.

The normal operation, lost frame, lost acknowledgment are depicted in the figures below.

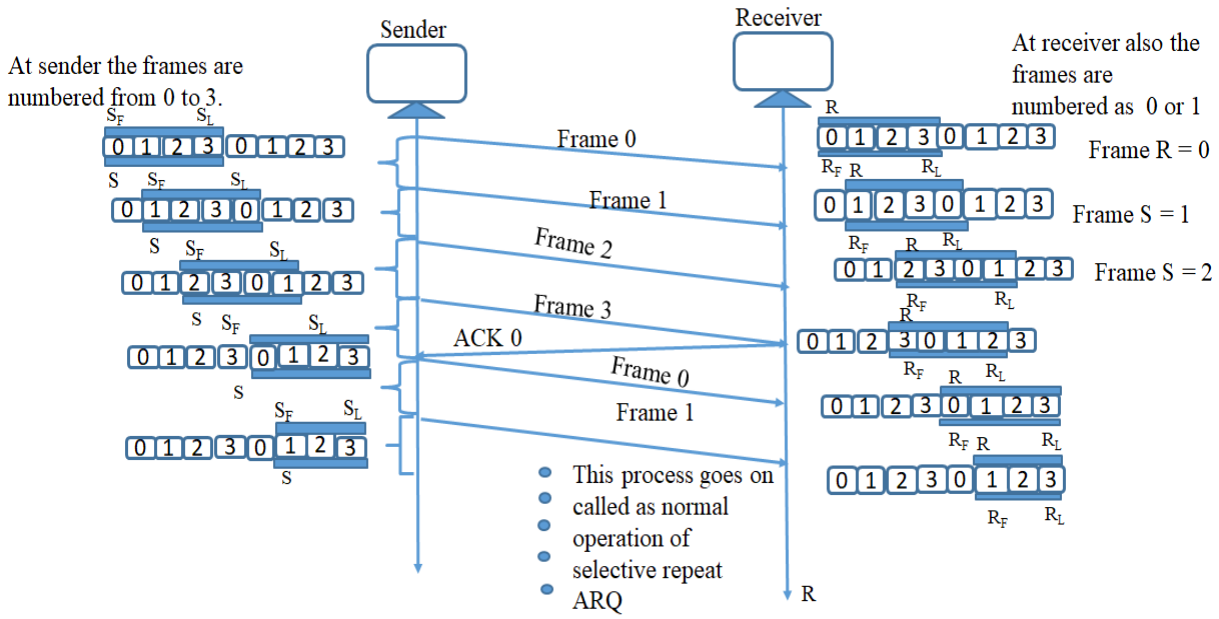


Fig.3.31. Normal operation for selective repeat ARQ.

3.4.4 Piggybacking

Stop and wit mechanism discussed above is seen in unidirectional transmission. However, we can have bidirectional transmission if the 2 parties (Sender and receiver) have two separate

channels for full duplex transmission or share the same channel for half duplex transmission. In this case each party needs both S and R (Window) variables to track frames sent and expected. Show Piggybacking provides better utilization of bandwidth in such cases. Acknowledgments are delayed until the next data frame is available for transmission. Acknowledgment will be hooked onto the outgoing data frame. Aata frame consists of an acknowledgment field whose size is of only few bits while acknowledgment frame comprises of several bits. Thus efficient use of bandwidth for transmission is used

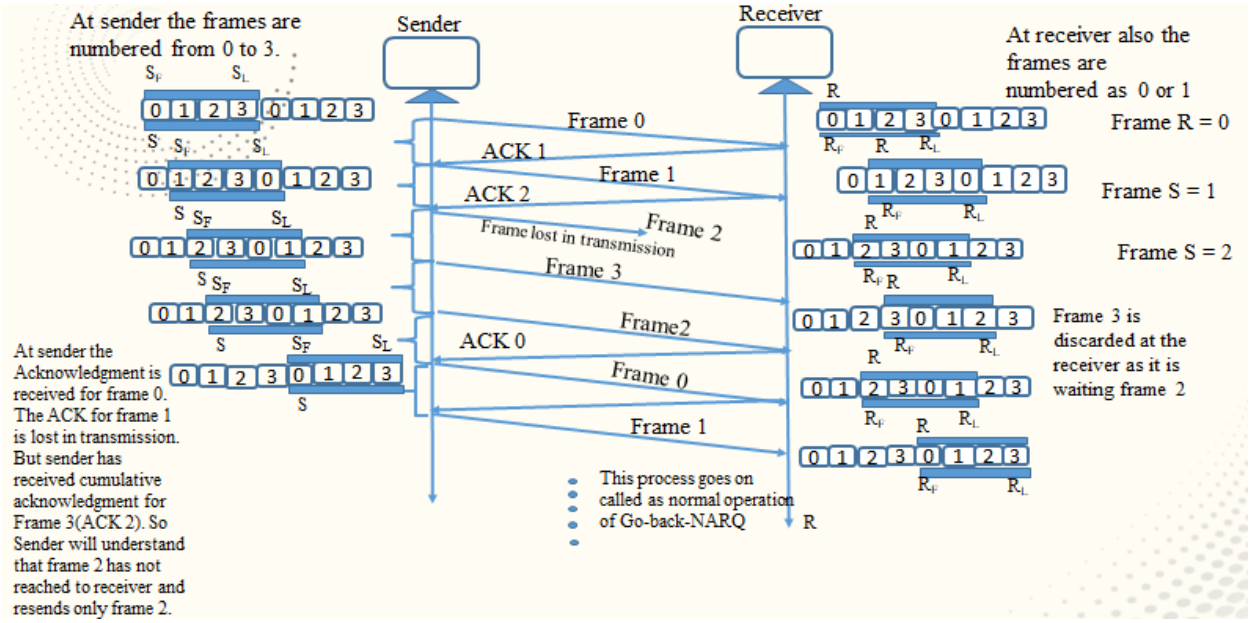


Fig.3.32. Lost Frame operation for selective repeat ARQ.

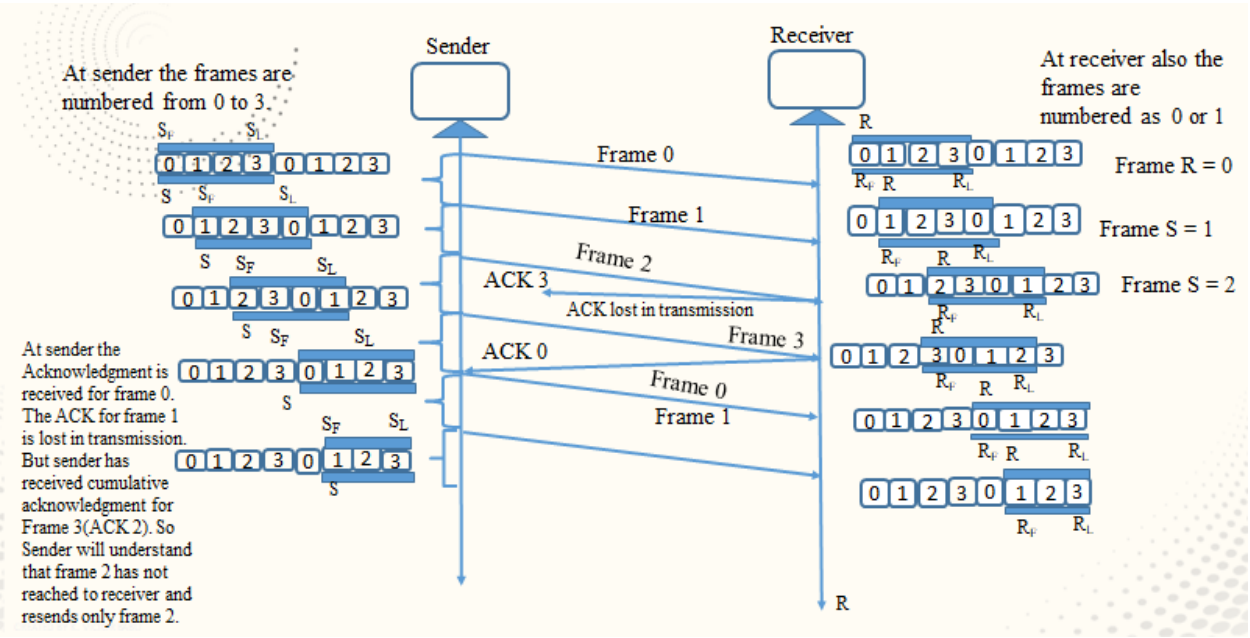


Fig.3.33. Cumulative acknowledgment Frame operation for selective repeat ARQ.

In selective repeat request the protocol will not repeat all the frames from the last acknowledgment. It will check the acknowledgments and retransmit only the frame for which it does not get an acknowledgment. So, this protocol is more efficient than Go-Back-N protocol.

Working Principe

1. Let us suppose we have two communication stations A and B.
2. The data flames transmitted have an acknowledgment field ack of few bits' length.
3. Additionally we have frames for sending acknowledgments (ack frames).
4. If station A wants to send both data and acknowledgement, it sends a data frame with ack field containing a sequence number of the frame to be acknowledged.
5. If station A wants to send only an acknowledgment, it waits for a finite time to check whether any data frame is available to send or not.
6. If the data frame is available then it piggybacks the acknowledgement with it, otherwise it sends ACK frame.
7. If station A has only a data frame to send it adds the last acknowledgement with it.
8. The station B discards all duplicate acknowledgments alternatively the station A may send data frame with the acknowledgment field containing a bit combination denoting no acknowledgement.

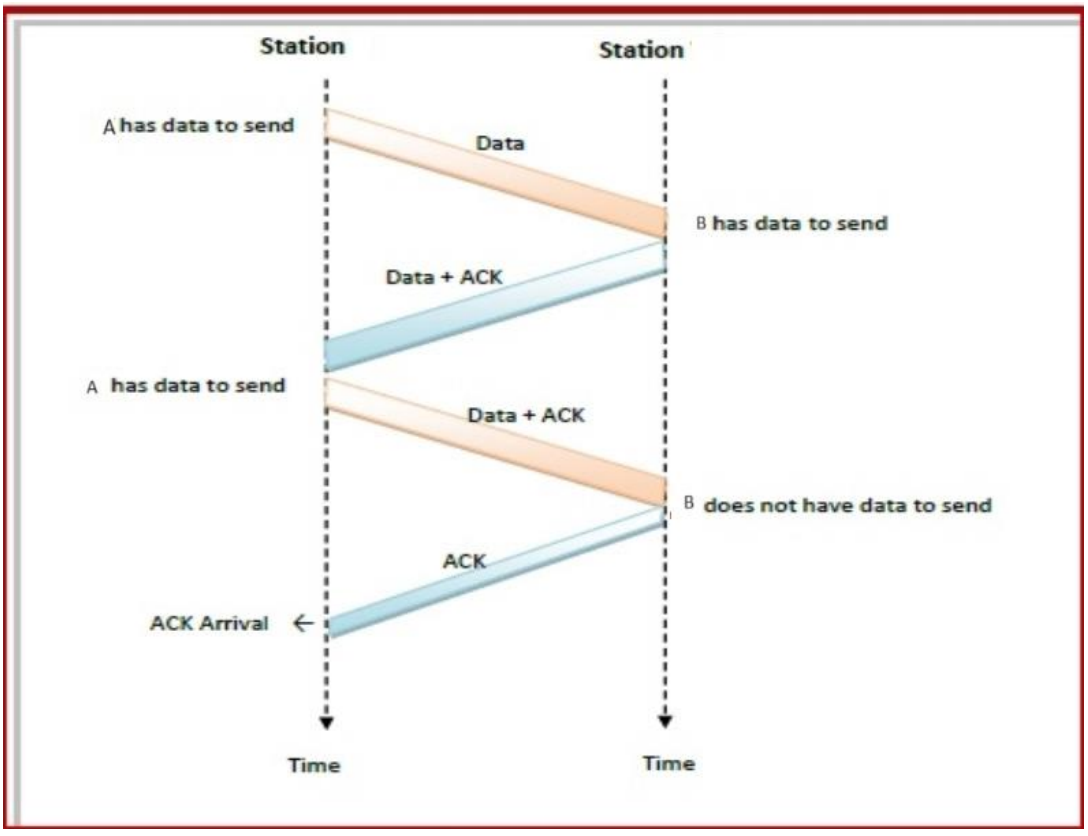


Fig.3.34. Piggybacking.

3.5 Medium Access Control

Data link layer in OSI model consists of two sub-layers called as link layer (LLC) and medium access control layer (MAC). Link layer controls the flow of frames between the sender and the receiver. It takes care of synchronization of the frames, detect errors at the receiver and tries to correct them. The transmitting devices are also called stations. All stations have equal priority to send and receive the inforamtion. Many protocols have been defined for doing this process such that no two stations speak at a same time, do not interrupt each other, do not monopolize the discussion and so on. These protocols are divided into three groups called as random-access protocols-controlled access protocols and channelization protocols. They're as shown Fig.3.35.

In random access or connection less methods no station is superior to other station, and none is assigned to control over another. At each instance a station that has data to send us a procedure defined by protocol to make additional on whether to send or not to send the data. In random access control we have protocols such as ALOHA, carrier sense multiple access (CSMA), carrier sense multiple access – collision detection (CSMA/CD), carrier sense multiple access – collision Avoidance (CSMA/CA). Here we are discussing only random access protocols.

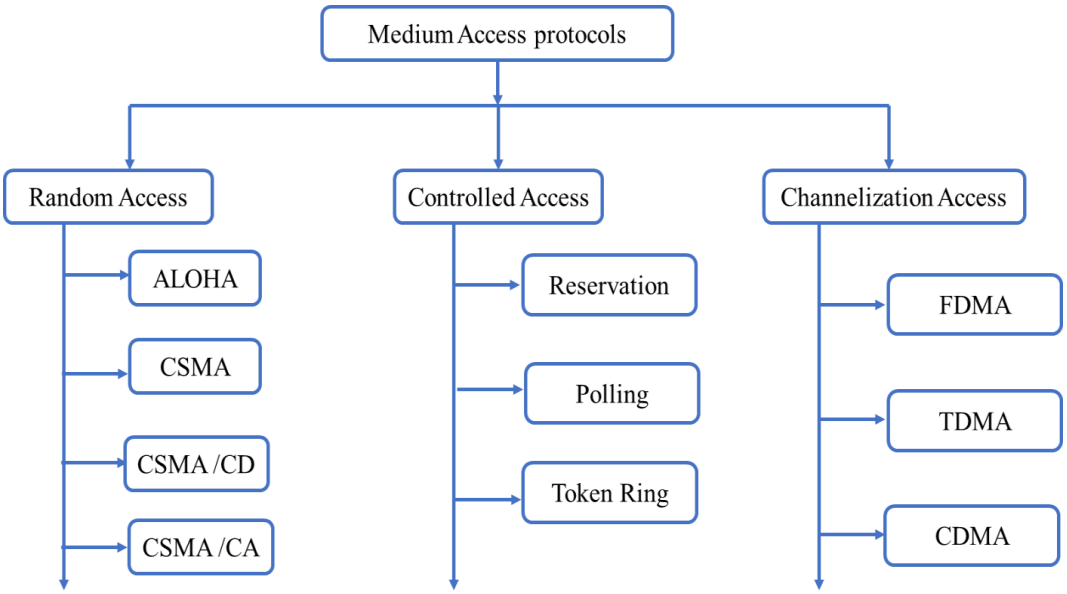


Fig.3.35. Medium access protocols.

3.5.1 ALOHA

1. Developed at University of Hawaii at early 1970's.
2. Design for radio LAN (Wireless communication).
3. It has potential collisions in the arrangement. The medium is shared between the stations.
4. It is a simple and elegant protocol.

5. Here the station sends the data whenever it has a frame to send, and we have only one channel to share between all the stations.
6. So there is a possibility of collisions between the data sent by the stations.
7. This method relies on acknowledgements from the receiver.
8. A collision involves two or more stations. If all these stations try to resend their frames after timeout, the frames will collide again.

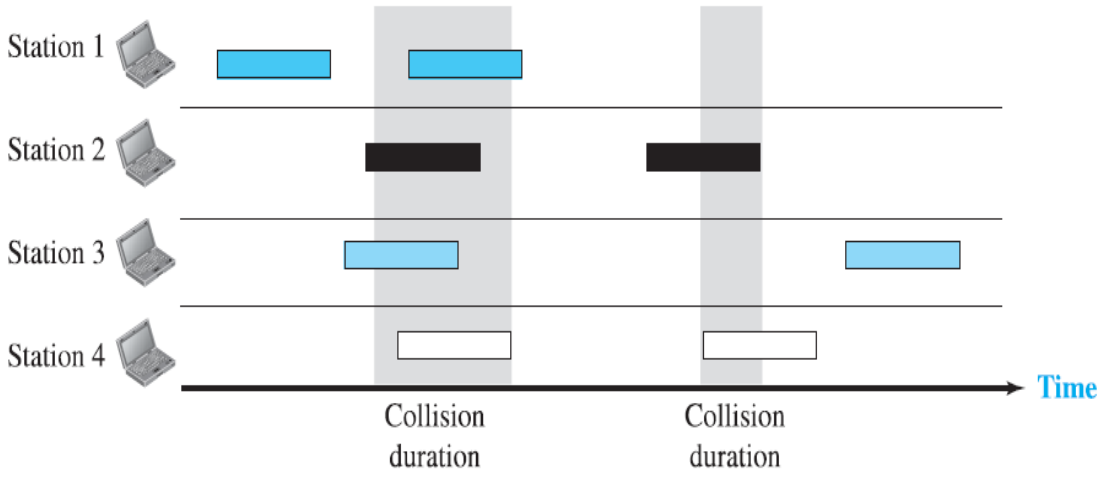


Fig.3.33. Collision in ALOHA.

9. Pure Aloha dictates that the station must wait for random amount of time before the timeout passes. This random time is called as back of time.
10. In pure ALOHA a second method is there to prevent congestion in the channel caused due to retransmitted frames.
11. After a maximum number of attempts K_{max} usually K_{max} = 15, station must give up and try later.
12. The time out period is equal to possible round trip transmit time ($2T_p$) to send a frame between most widely separated stations.
13. The back of time T_B is a random value that normally depends on the K_{max}, (number of attempts of transmission).
14. In this method for each transmission a multiplier R=0 to 2^K-1 is randomly chosen and multiplied by T_p or T_{fr} – the average time to send out a frame.
15. The value of K_{max} is usually chosen a s15.

In the below figure K = number of attempts

T_p = Maximum propagation time

T_{fr} = Average transmission time

T_B = Back off time = Rx T_p

R = random number between 0 to 2^K-1

Vulnerable time

1. This is a time where there is a possibility of collision.
2. Let us assume the stations are sending fixed length frames taking T_{fr} seconds to send.

3. Let us assume that station B sends a frame at time t. Now imagine the station here started to send its frame at t - T_p. This leads to collision between frames from station B station A.
4. On the other hand suppose that station C starts to send a frame at t + T_{fr}.
5. Here there is also a collision between frames of Station B and C.
6. So vulnerable time = $2T_{fr}$.

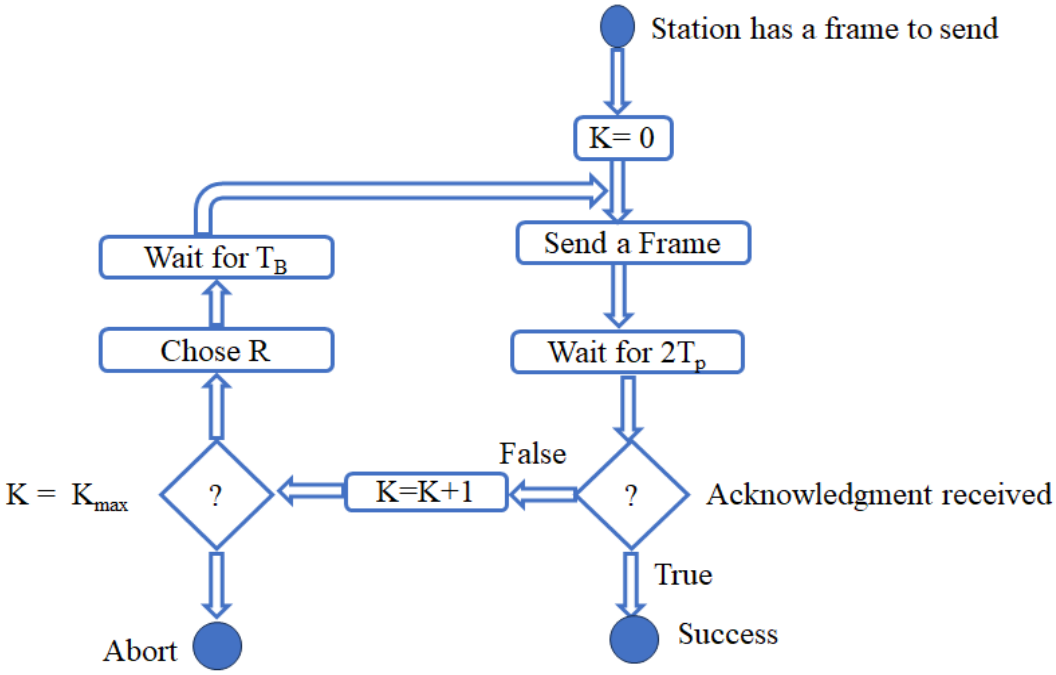


Fig.3.34. Flow chart of ALOHA.

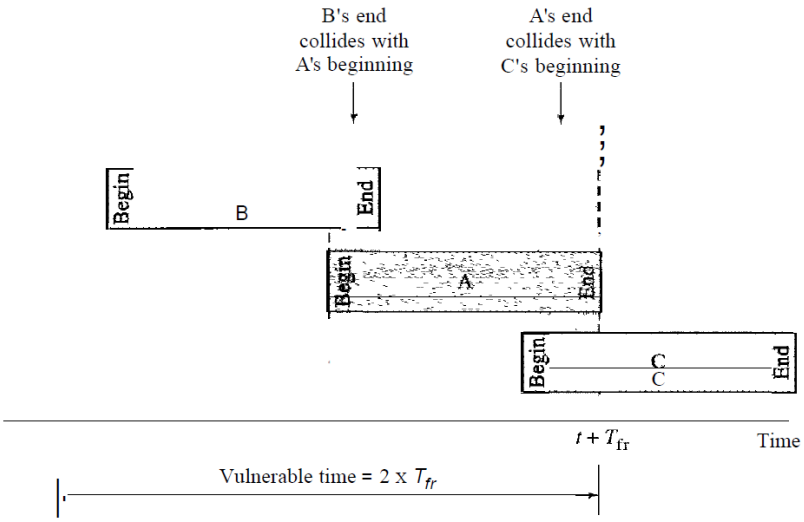


Fig.3.35. Vulnerable time.

Example: A pure Aloha Network transmits 150 bits for frame on a second channel of 250 kbps what is the requirement to make the channel collision free?
Sol: Average transmission time = 150/250kbps = 0.6ms. So vulnerable time = 2*0.6ms = 1.2ms.
This means that no station should start transmission 1.2ms before this station starts transmission and no station should start transmission during this time.

Throughput:

1. Let us assume that there are infinite number of stations are generating frames based on Poison distribution with mean N frames per frame time.
2. In N>1 then the stations are generating frames at a rate that is greater than the channel capacity and therefore nearly every frame will suffer a collision.
3. Let us assume the probability of K transmission attempts for frame time, old & new combined are also in Poison distribution with a mean of G frames per frame time. At low load on the line then G = N and at high load G > N (we will have many collisions).
4. In all load conditions throughput S = offered load G * probability P_0, where P_0 is the probability of successful transmission i.e., frame not suffering from collision. So, S = G P_0.
5. Now probability of m frames generated during a given frame time is

$$P_r = \frac{G^m e^{-G}}{m!}$$

6. So the probability of zero frames is

$$P_r = \frac{G^0 e^{-G}}{0!}$$

7. So, all frames suffer from collision m = 0. Now, in a interval of $2T_{fr}$ long the mean number of frames generated is 2G.
8. So, the probability that other traffic generated during the entire vulnerable time is thus given by $P_0 = e^{-2G}$. So S = G* P_0 or S = G* e^{-2G}.
9. The maximum throughput occurs at G = 0.5 and the value of throughput is 0.18 i.e., 82% of frames end up in collisions and are therefore lost in transmission.

Advantages of Pure ALOHA

1. Simple and easy to implement protocol.
2. Random access and uncoordinated transmission.
3. Continuous attempt to reach destination even collision has occurred.
4. Low overhead and acknowledgment based.
5. Predictable access time and LAN based protocol.

Disadvantages of Pure ALOHA

1. High collision rate
2. Low throughput
3. lack of scalability
4. Inadequate collision resolution.

3.5.2 Slotted ALOHA

1. Pure Aloha is a vulnerable time of two into 2* T_p.
2. Here the station may send some soon after another station has started or just before railway station has finished. Slotted Aloha was invented to improve the efficiency of pure aloha.
3. In slotted aloha we divide the time into slots of T_{fr} seconds and force the station to send only at the beginning of the time slot.
4. Now a station is allowed to send only at the beginning of the synchronised time slot. If the station loses any time slot it must wait until the beginning of the next time slot.
5. Of course, there is still the probability of collision of two stations trying to send at the beginning of the same time slot.
6. However, the vulnerable time is now reduced to one half equal to T_p.

Throughput

1. In slotted ALOHA vulnerable time is $2T_{fr}$.
2. The average number of successful transmissions is S = G*exp(-G).
3. So the maximum throughput is S_{max} is 0.368, when G =1.
4. In slotted ALOHA case the 36.8% of the frames reach destination successfully.

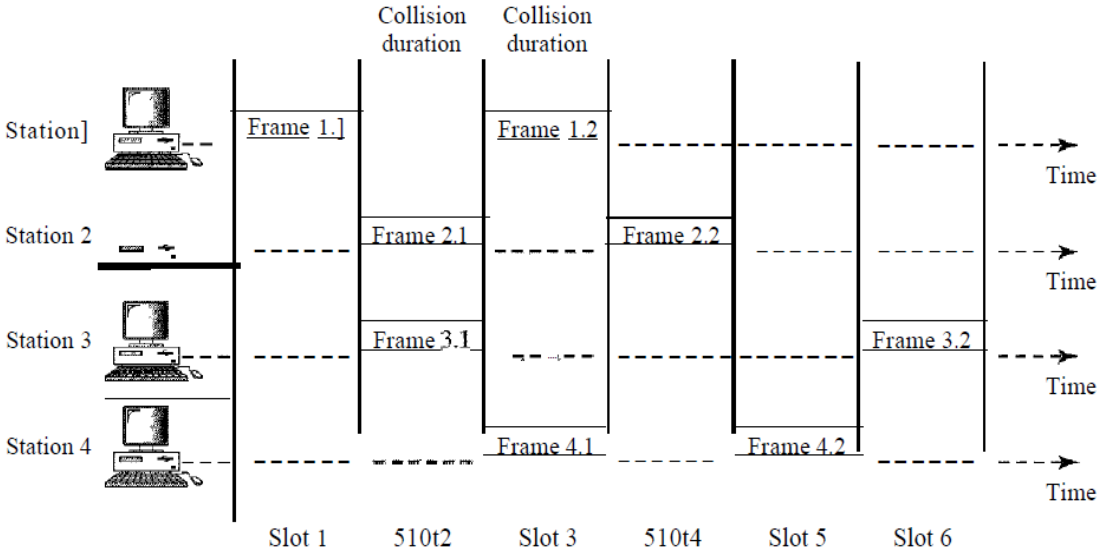


Fig.3.36. Time slots in slotted ALOHA

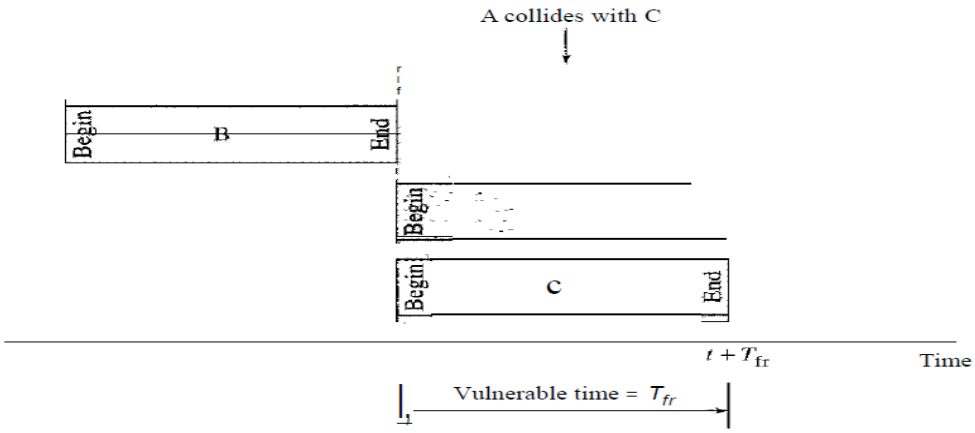


Fig.3.37. Vulnerable time in slotted ALOHA.

3.5.3 Carrier Sense Multiple Access

1. To minimise the chance of collision and to increase the performance carrier sence multiple access method was developed.
2. The chance of collision can be reduced if the station senses the medium before trying to use it.
3. Here we reduce the probability of a collisions, but we cannot eliminate it.
4. The probability of collision still exists because of propagation delay when station sunset frame it still takes time for the first bid to reach every station and for every station to sense it.
5. So as the first bit sent by station has not yet received the receiver another station will sense the carrier is free.

Vulnerable time

1. The vulnerable time for CSMA is propagation time Tp.
2. Time needed for the signal to propagate from 1 end of the medium to other end is called vulnerable time.
3. So here once the receiver senses the first bit of the frame then remaining stations will hear the bit and will not transmit.

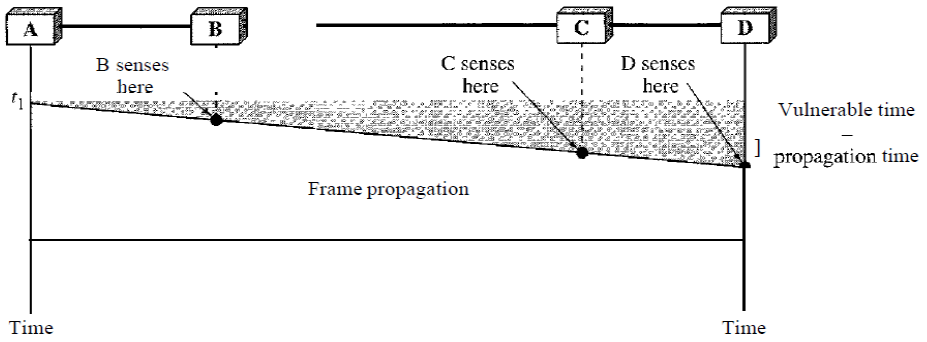


Fig.3.

38. Vulnerable time in CSMA.

Persistence Methods

1. Whenever the station finds the channel is busy it will wait for some time.
2. It then again cheques whether the channel is free or not and then transmits the data.
3. In this there are three methods.
   a. I Persistent
   b. non-persistent
   c. P - persistent

I – Persistent

1. This is a simple method.
2. Here the station senses the channel and if it is free, it immediately transmits the data.
3. Here there is a chance of other stations also detecting the channel is free and transmitting the data, so collisions are more here. This method is used in Ethernet.

Non persistent

1. In non-persistent method once the station finds the channel is idle it sends a frame, and it finds the channel is busy it waits for some random time.
2. After the random time is completed it again senses the channel and tries to retransmit the frames.

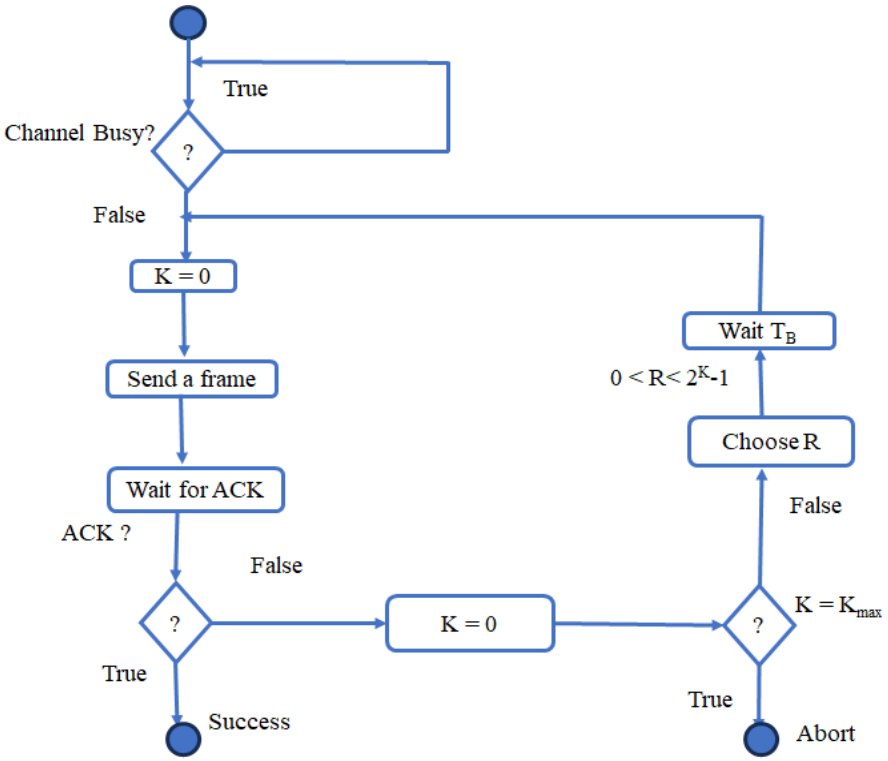


Fig.3.39. Flow chart of I-persistent.

3. In this method the chances of collisions have been reduced but here two or ore stations may wait for some amount of random time and retry for transmission.
4. So it reduces the efficiency of the network because the medium remains idle when there may be station with frames to send.

Advantages of I- persistent methods

1. Deterministic approach – more likely to transmit once the channel is idle.
2. Simple to implement.
3. Straight forward method to implement.

Disadvantages of I-persistent methods

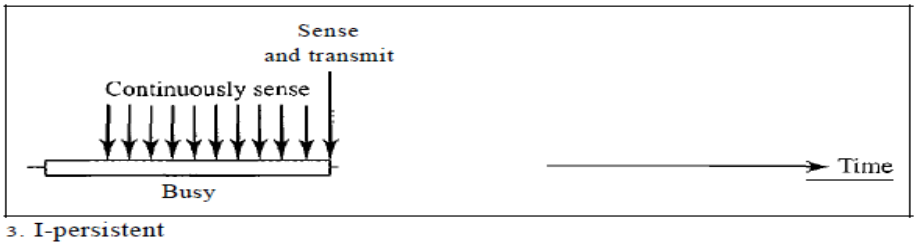
1. Collision rate is high.
2. Unfair method where all the devices are not given a same priority.

Advantages of P- persistent methods

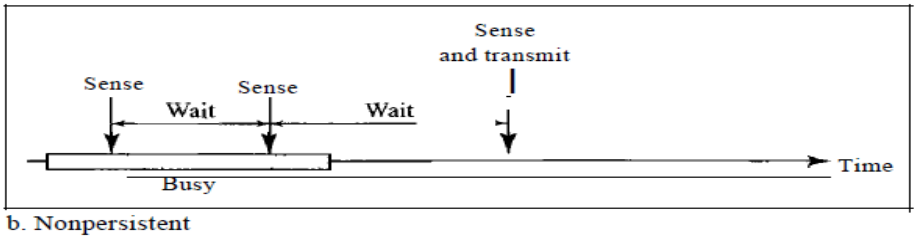
1. Collision avoidance is improved.
2. Fairness in accessing the channel by another device is implemented.

Disadvantages of P-persistent methods

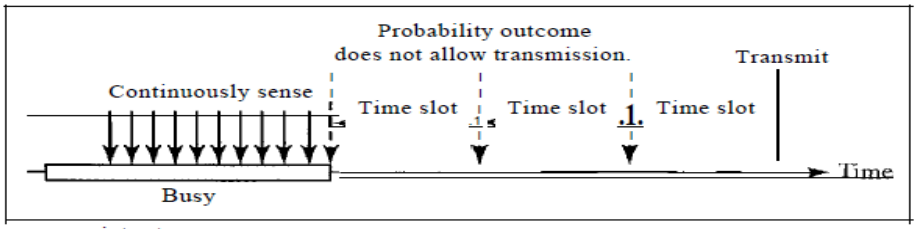
1. Complex to implement.
2. If p value is high then latency is more.



3. I-persistent



b. Nonpersistent



c. p-persistent

Fig.3.40. Flow diagrams for three persistent methods.

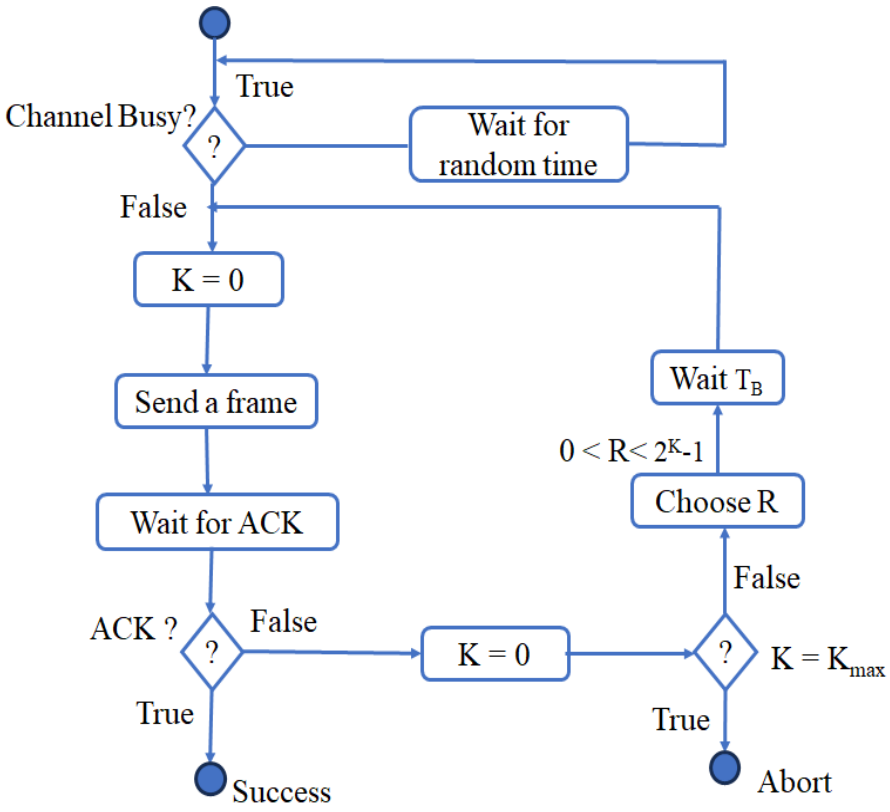


Fig.3.40. Flow chart for N - persistent method.

3. P - persistent method

1. In P persistent mothed the channel has time slots with the slot duration equal to or greater than the maximum propagation time.
2. It combines the advantages of other two strategies.
3. It reduces the chance of collision and improves efficiency.

Here the station follows these steps to find the line is ideal or not.

Step 1: With probability P the station sends frame.

Step 2: With probability 1 – P = q the station waits for the next time slot and checks the line for transmission.

Step 3: If the line is idle it goes to step one.

Step 4: If the line is busy, it confirms that a collision has occurred and uses back off procedure.

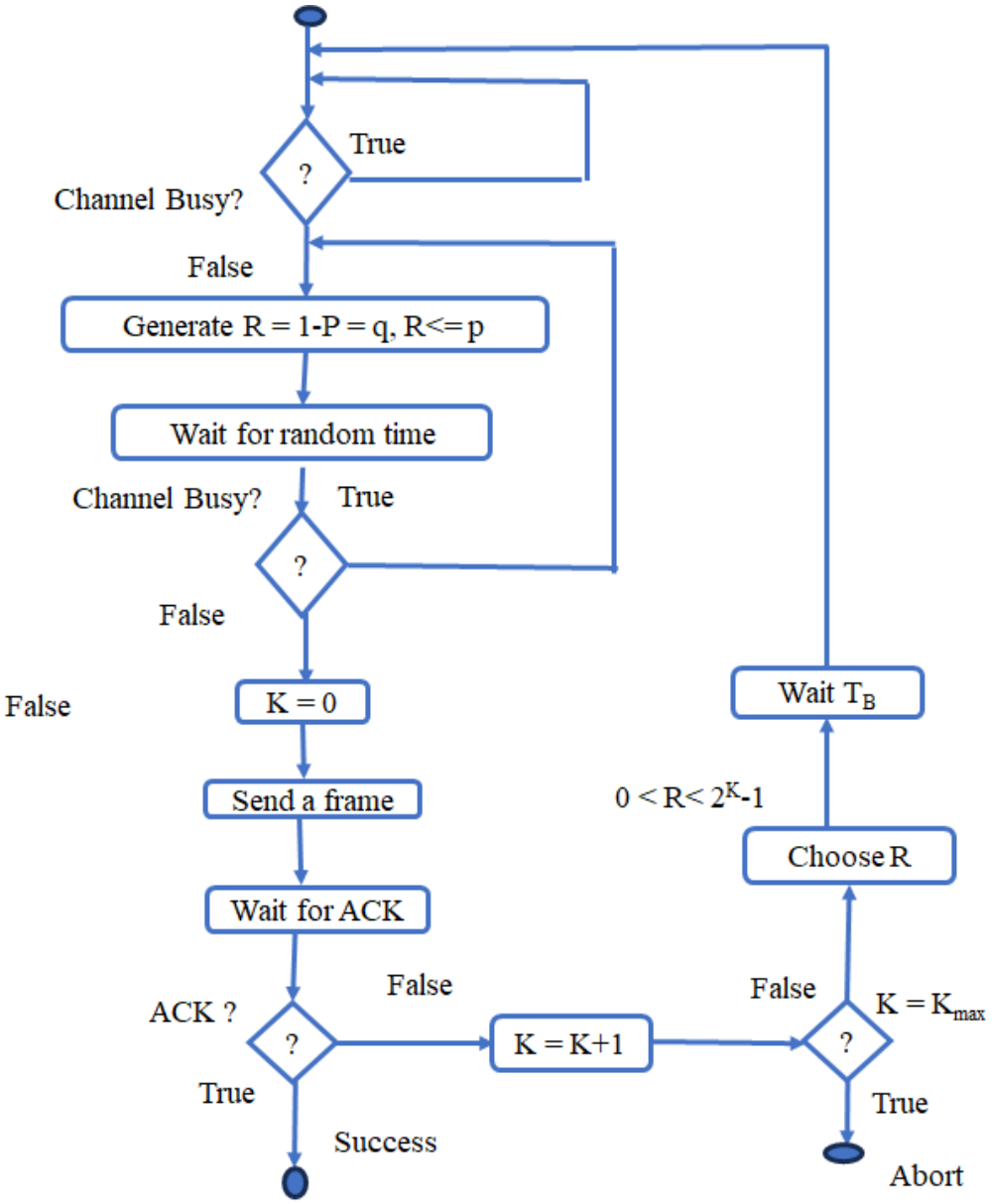


Fig.3.41. Flow chart for P - persistent method.

Pure ALOHA	Slotted ALOHA
Any station can transmit at any time – Random access	Stations can send in the given time slot only.
Time is continuous and not synchronized globally	Time is discreate and globally synchronized
Vulnerable time is $2*T_p$.	Vulnerable time is T_p.
Probability of successful transmission of the data packet = $G \times e^{-2G}$	Probability of successful transmission of the data packet = $G \times e^{-G}$