2. Data Link Layer.

→

Data Link Layer Design Issues:-

→ The data link layer has a no. of specific functions to carried out.

→ These functions include Providing a well-defined service interface to the network layer, determines how the bits of physical layer are grouped into frames, dealing with transmission Errors & regulating the flow of frames so that slow receivers are not swapped by fast senders.

→ The following are the Design issues of data link layer.

1, Services Provided to the Network layer
2, Framing
3, Error Control
4, Flow Control.

1, Services Provided to the Network layer:-

→ The Principal Service is to transfer data from the network layer of on the source machine to the network layer of the destination machine.

→ Services offered by ~~the data~~ can vary from System to System. Three reasonable possibilities that are commonly Provided are:

ⓐ Unacknowledged Connectionless Services
ⓑ Acknowledged Connectionless Services
ⓒ Acknowledged Connection-Oriented Services.

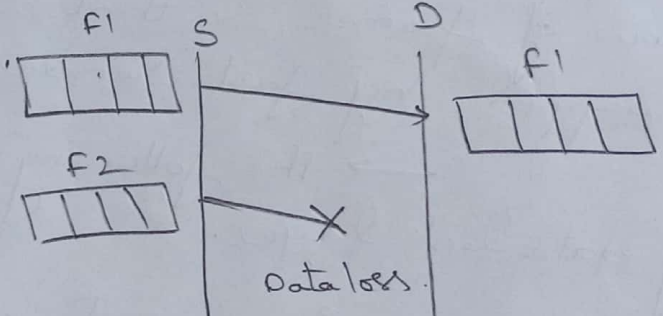
ⓐ Unacknowledged Connectionless Services:-

→ It consists of having Source machine Send independent frames to the destination machine without having the destination acknowledge them.

→ No Connection is Established beforehand or released afterward.

→ If a frame is lost due to noise on the line, no attempt is made to recover it in the data layer.

Eg:- Speech.



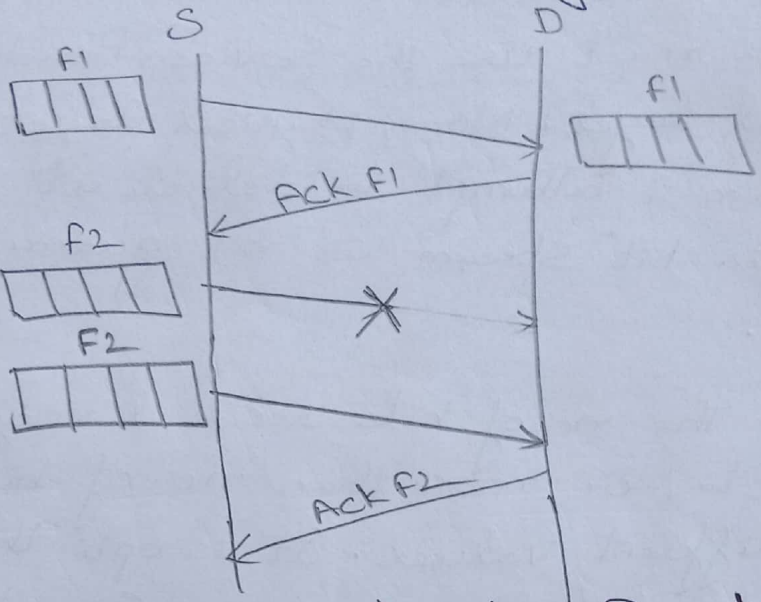
ⓑ Acknowledged Connection-less Services:-

→ Here Service is offered, there are Still no Connections used, but Each frame ~~is~~ Sent is individually acknowledged.

→ So, that Sender knows whether or not a frame has arrived Safely.

→ If it ~~is~~ has not arrived within a Specified time interval, it can be Sent again.

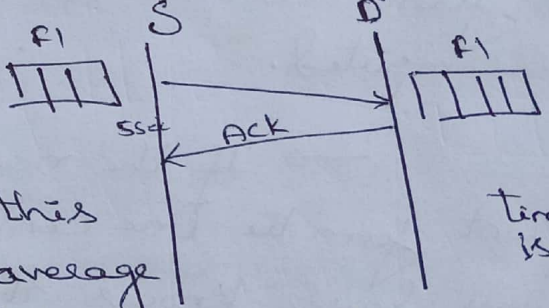
→ This Service is Useful Over Unreliable channels, such as wireless systems.



© Acknowledged Connection-~~less~~ Oriented Services:-

→ The transport layer can always send a message & wait for it to be acknowledged.

→ If the acknowledgement is not forthcoming before the timer goes off, the sender can just send the entire message again.



→ ~~The~~

→ the trouble with this strategy is that if the average message is broken up into say 10 frames, & 20% of all frames are ~~are~~ lost, it may take very long time for the message to get through.

→ If individual frames are acknowledged ~~by~~ & re-transmitted, entire message get through much faster.

→

(5.) Framing :-

→ To Provide Service to the network layer, the data link layer must use the Service Provided to it by the physical layer. Now, physical layer accepts raw bits stream & attempt to deliver it to the destination. This bit Stream is not guaranteed to be Error free.

→ The no. of bits received may be less than or Equal to, or more than no. of bits transmitted, & they have different values. It is upto the datalink layer to detect, & if neccessary, Correct Errors.

→ The Usual approach in the & datalink layer to break the bit stream up into discrete frames & Compute the checksum for Each frame. When frame arrives at the destination, the checksum is recomputed.

→ If the newly Computed checksum is different from the One Contained in the frame, the data link layer knows that an Error has occured & take steps to deal with it (Eg: discards the bad frame & Send back an Error report).

→ One way to achieve this framing is to insert time gaps between frames (i.e like Spaces between words in a text). Since it is too risky to Count on timing to mark the Start & End of Each frame, four methods have been devised. They are:

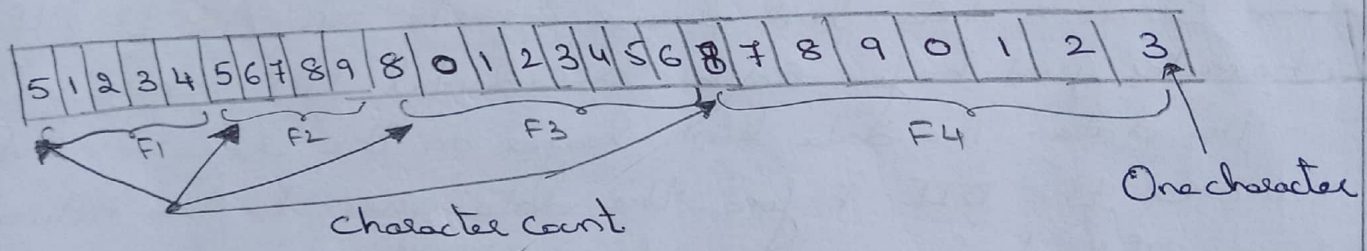
(a) Character Count
(b) Starting & Ending characters with Character stuffing
(c) Starting & Ending flags with bit stuffing
(d) physical layer coding violations.

(a) Character Count:-

→ It uses a field in the header to specify the number of characters in the frame.

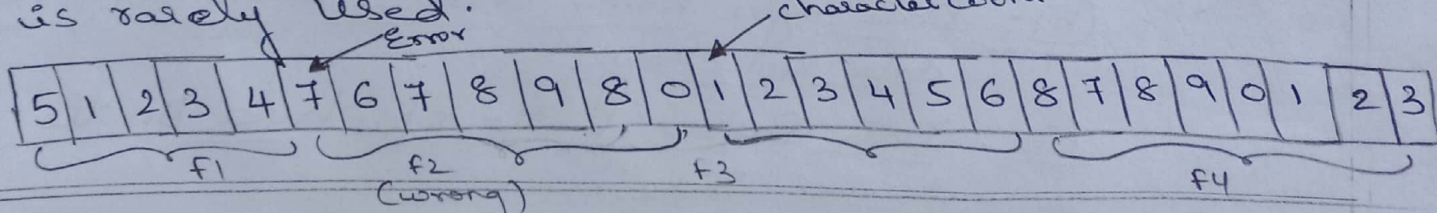
→ When the data link layer at the destination sees the character count, it knows how many characters follow, & hence where the end of the frame is.

→ for Example, the below fig(a) has four frames with sizes 5, 5, 8, & 9 characters respectively.



fig(a): A character stream without Errors.

→ for Example, in above fig(a), if the character count of 5 in the second frame (F2) becomes 7 (shown in fig(b)), the destination will out of Synchronization & will be Unable to locate the start of the next frame. Retransmitting is not Possible because the starting of the characters is Unknown. for, this reason, the character count method is rarely used.



fig(b): A character stream with one Error

⑥ Starting & Ending characters, with Character Stuffing:-

→ To Overcome the above Problem, for Each frame ASCII character DLE STX (Datalink Estape, Start text) is added at the starting of the frame & Similarly DLE ETX (Data link Escape End of the text) is added at the End of the flame.

→ A Serious Problem occurs with this method when binary data, such as object Programs or floating-Point numbers are being transmitted.

→ It is Easily applicable for characters. ~~the Sender at the datalink layer inserts DLE~~ for DLE STX & Or DLE ETX occurs in the data, which interfere with the framing.

→ One way to Solve this Problem is to have Sender's data link layer inserts an ASCII DLE character just befor Each "accidental" DLE character in the data. The data link layer on the receiver End removes the DLE before the data given to the network layer. This technique is ~~that~~ called character stuffing.

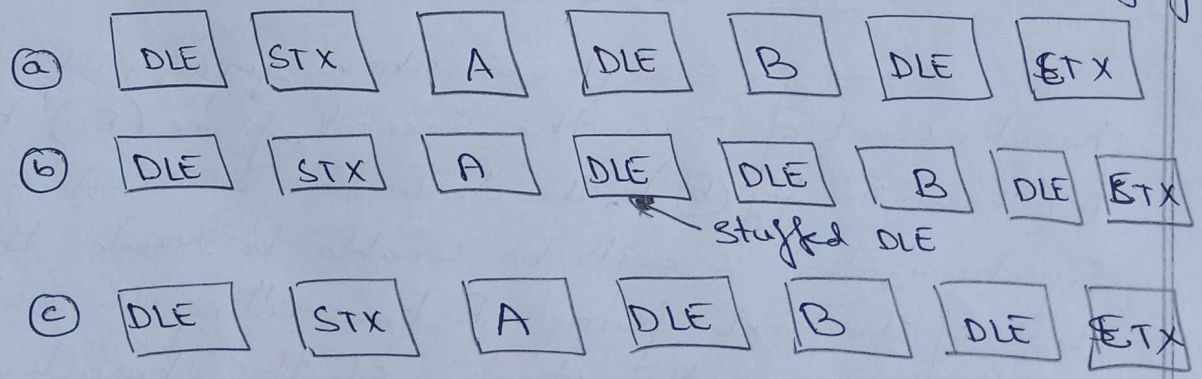


fig ② ⓐ: Data Sent by the n/w layer

ⓑ: Data after being characters stuffed by the data link layer.

ⓒ: Data Passed to the n/w layer On the receiver side.

→ A major disadvantage of Using this framing method is that it is closely tied to 8-bit characters in general & the ASCII character code in Particular. As networks developed, the disadvantage of Embedding the character code in the framing mechanism became more & more obvious. So now

© Starting & Ending flags with bit Stuffing:-

→ To overcome above method, a new technique allows data frames to Contain an arbitrary number of bits & allows character codes with an arbitrary no. of bits Per character.

→ Each frame begins & Ends with a Special bit Pattern 01111110 called a flag byte.

→ When Sender's data link layer Encounters five ~~Consecq~~ Consecutive Ones in the data, it automatically stuff's a 0 bit into the Outgoing bit Stream. This technique is called bit Stuffing.

→ Where, if the receive Sees five Consecutive incoming 1 bits, followed by 0-bit it automatically destuffs the 0-bit.

Eg:- flag Pattern → 01111110

Sender → 011111010

Receiver → 01111110

→ the disadvantage of bit-Stuffing is that the boundary between two frames can be unambiguously recognized by the flag Pattern.

→ Thus receiver looses track of where it is, all it has to do is Scan the input for flag Sequences, Since they can only occur at the frame boundaries & never within the data.

(d) Physical layer Coding Violations :-

→ It is applicable for network in which the Encoding on the physical medium Contains some redundancy.

→ for Example, some LAN's Encode 1-bit of data by using 2-physical bits. Generally 1-bit is a high-low pair & 0-bit is a low-high pair. The Combination of high-high & low-low are not used for data.

→ The Scheme means that Every data bit has a transition in the middle, make it Easy for the receiver to locate the bit boundaries.

(3) Error Control :-

→ Error Control includes both Error detection & Error Correction.

→ It allows the receiver to inform the Sender if a frame is lost or damaged during transmission & Coordinates the retransmission of those frames by the Sender.

→ Error Control in this layer is based on Automatic Repeat Request (ARQ). When Error detected Specified frame is re-transmitted.

④ Flow Control:-

→ It coordinates the amount of data that can be sent before receiving acknowledgement.

→ It is a set of Procedures that tells the sender how much data it can transmit before it must wait for an acknowledgement from the receiver.

→ Receiver has a limited speed at which it can process incoming data & a limited amount of memory in which to store incoming data.

→ Receiver must inform the sender before the limits are reached & request that the transmitter to send fewer frames or stop temporarily.

→ Since the rate of processing is often slower than the rate of transmission, receiver has a block of memory (buffer) for storing incoming data until they are processed.

2. Error Detection & Correction:-

→ Telephone System has three Parts: the switches, the interoffice trunks & the local loops.

→ first two are now almost entirely digital in most developed countries.

→ the local loops are still analog twisted copper pairs & will continue to be so for years due to Enormous Expense of replacing them.

@ Error-Detecting Codes:-

→ Whenever a message is transmitted, it may get scrambled by noise or data may get corrupted

→ To avoid this, we use Error-detecting Codes which are additional data added to a given digital message to help us detect if any Error has occurred during transmission of the message.

→ Some Popular techniques for Error detection are:

1. Simple-Parity check
2. Two-dimensional Parity check
3. Check-Sum
4. Cyclic redundancy check.

1. Simple-Parity check:-

→ Block of data from the source are subjected to check bit or Parity bit, where a Parity of:

- 1 is added to the block if it contains odd no of 1's Ex
- 0 is added if it Contains Even number of 1's.

→ This Scheme makes the total no. of Even 1's, i.e why it is called Even Parity checking.

Ex:- 100011

↓

1000111

(b) Two-dimensional Parity check:-

→ Parity check bits are calculated for each row which is equivalent to simple parity check bit.

Eg:-

10011001	11100010	00100100	10000100

Row Priorities.

10011001	0
11100010	0
00100100	0
10000100	0
11011011	0

Column Priorities →

100110010	111000100	001001000	100001000	110110110

Data to be sent.

(c) Check Sum:-

→ Here, the data is divided into k-segments each of m-bits

→ At the sender side, the segments are adding using 1's complement arithematic to get the sum. The sum is complemented to get the checksum.

→ The check sum segment is sent along with the data segments.

→ At the receiver's end, all the received segments are added using 1's complement arithematic to get sum. The sum is complemented.

→ If result is zero, the received data is accepted, otherwise discarded.

Original Data

10011001	11100010	00100100	10000100
1	2	3	4

K=4, m=8.

Sender

1. 10011001
2. 11100010

(1) 01111011
↳ 1

01111100

3. 00100100

10100000

4. 10000100

(1) 01001000
↳ 1

Sum: 00100101

Checksum: 11011010.

Receiver

1. 10011001
2. 11100010

(1) 01111011
↳ 1

01111100

3. 00100100

10100000

4. 10000100

(1) 00100100
↳ 1

00100101
11011010

Sum: 11111111
Comp: 00000000

Conclusion: Accepted Data

④ Cyclic Redundancy check (CRC):-

→ CRC is based on binary division.

→ In CRC, a sequence of redundant bits, called CRC are appended to the end of data unit so that the resulting data unit becomes Exactly divisible by a second, Pre-determined binary number.

→ At the destination, the incoming data unit is divided by the same number. If at this step there is no remainder, the data unit is divided by the same number. If at this step there is no remainder, the data unit is assumed to be Correct & is therefore accepted.

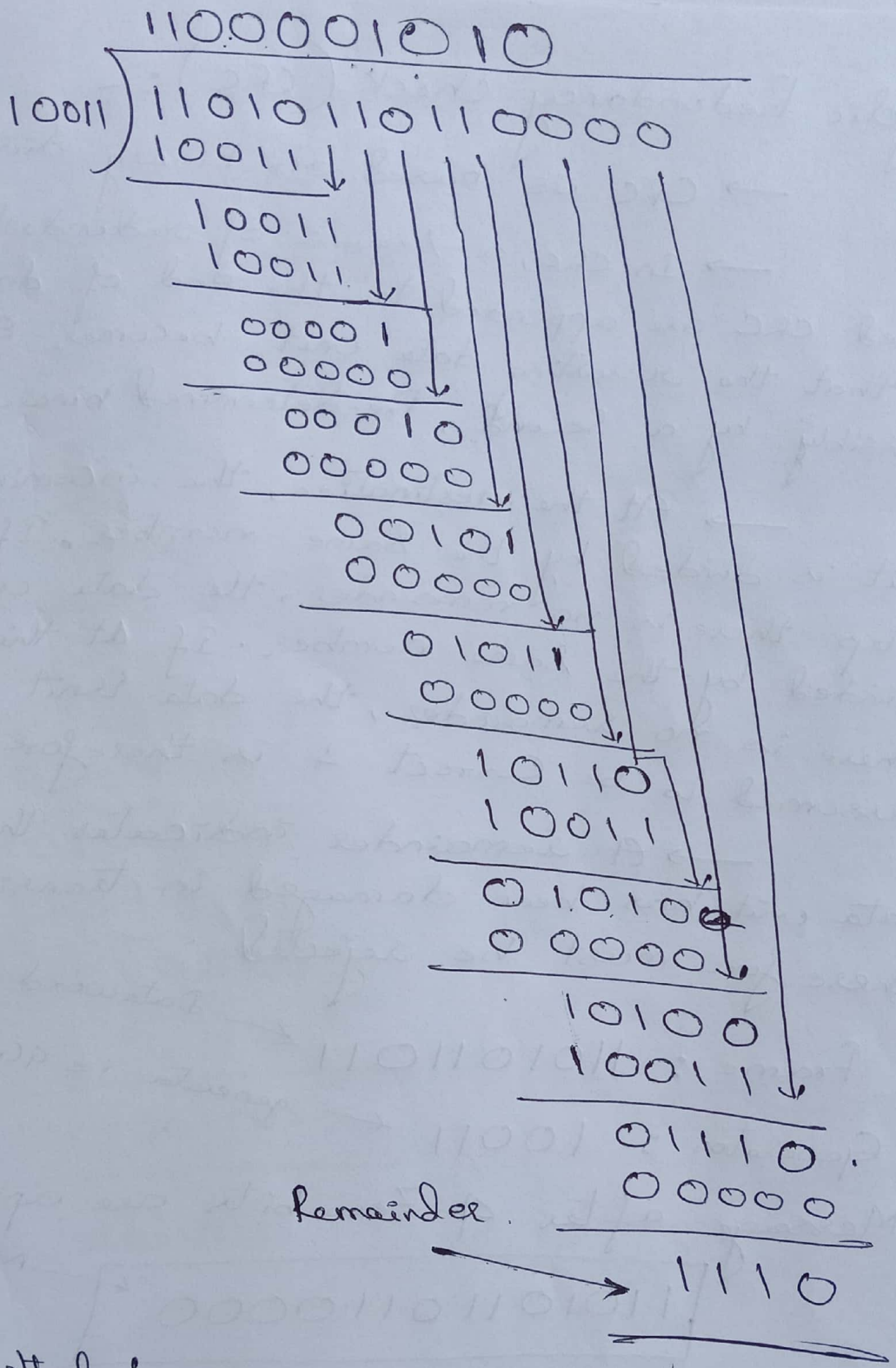
→ A remainder indicates that the data unit has been damaged in transit & therefore must be rejected.

Eg:- Frame : 1101011011 ← Dataword.

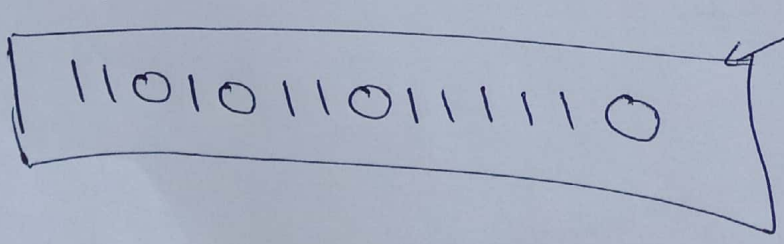
Generator : 10011 ← generator i.e G(x)

Message after 4 Zero bits are appended:

11010110110000 ← M(x)



Transmitted frame :-



MEDIUM ACCESS SUBLAYER.

Introduction:-

→ Data-link layer is divided into two sub-layer i.e into two functionality-Oriented sublayers. They are:-

(a) Data-link Control
(b) Multiple-access resolution.

→ The upper sublayer is responsible for flow & Error Control is called as Logical link Control (LLC) layer.

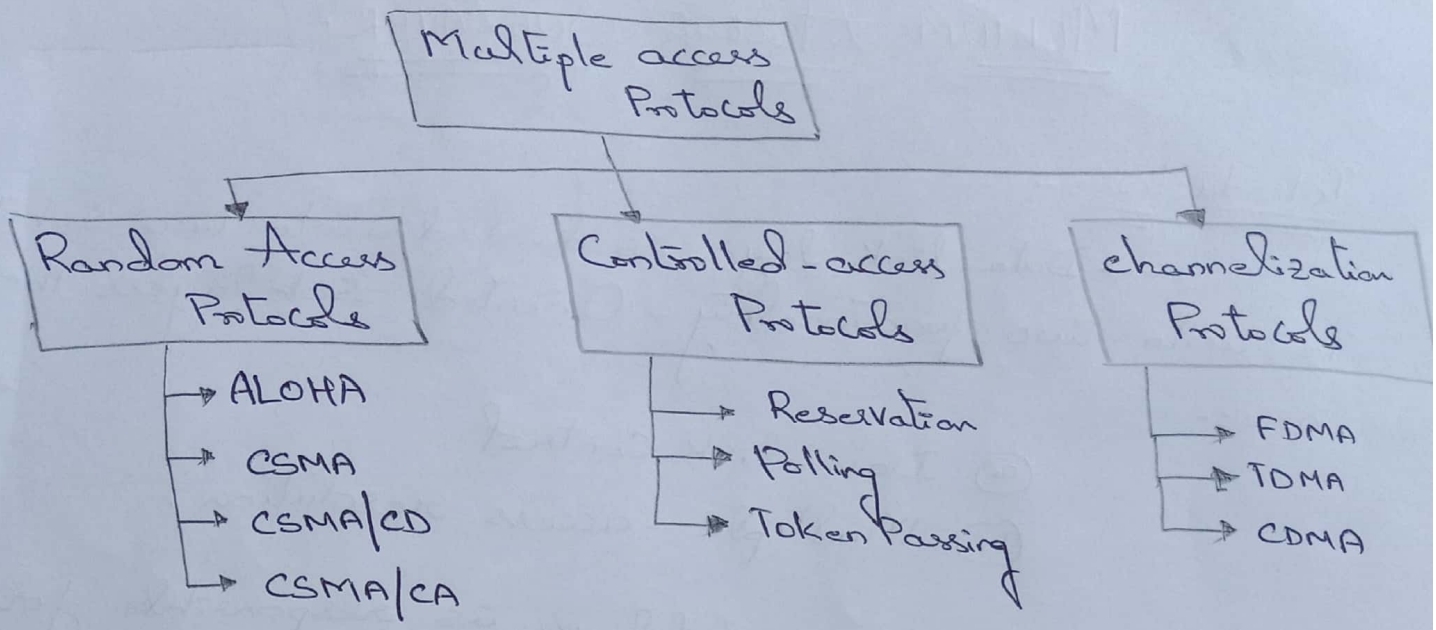
→ The lower sub-layer is responsible for multiple access resolution is called as Media access Control (MAC) layer.

→ When nodes or stations are Connected & Use a Common link, called a multi-point or broadcast link, we need a multiple-access Protocol to Coordinate access to the link.

→ The Problem of Controlling the access to the medium is similar to the rules of Speaking in an assembly.

Eg:- Conference call (i.e 5-People).
Meeting manually.

→ To Overcome this multi-point networks, many formal Protocols have been devised to handle access to shared link. These Protocols are Categorized into three groups. They are:-



1. RANDOM ACCESS:-

→ A Station that has data to Send uses a Procedure defined by the Protocol to make a decision on whether to Send or not.

→ The decision depends on the state of medium (idle or busy).

→

Features:-

1. No Scheduled time for a Station to transmit. Transmission is random among the Stations. So, it is called Random Access.
2. No rules Specify which Station should Send next. Station Compete with another to access the medium. So, it is called Contention methods.

→ In random access method, Each Station has the right to the medium without being Controlled by other Stations. However, if more than One Station tries to Send, there is an access Conflict occurs collision (i.e frame will be Either destroyed or modified).

→ To avoid access conflict or to resolve, each station follows a Procedure that answers the following questions:

1. When can the station access the medium?
2. What can the station do if the medium is busy?
3. How can the station determine the success or failure of the transmission?
4. What can the station do if there is an access conflict?

→ Random Access methods are classified into four types. They are:-

(a) ALOHA
(b) CSMA
(c) CSMA/CD
(d) CSMA/CA.

(a) ALOHA:-

→ ALOHA was developed at University of Hawaii in Early 1970, by Norman Abramson.

→ It is designed for radio, Wireless LAN, but it can be used on any shared medium.

→ When a station sends data, another station may attempt to do so at the same time. The data from the two stations collide & become garbled.

→ Pure & Slotted ALOHA are two Versions of ALOHA.

→ They differ with respect to whether or not time is divided up into discrete slots into which all frames must fit. Pure ALOHA doesn't require global time synchronization, but slotted ALOHA does.

1. Pure ALOHA:-

→ The Original ALOHA Protocol is called Pure ALOHA.

→ The idea is that Each Station Sends a frame whenever it has a frame to send.

→ However, since there is only one channel to share, there is the Possibility of collision between frames from different stations.

→ The below fig ⓐ, there are four stations that Contend with one another for access to the shared channel. In above fig ⓐ Each station sends two frames (i.e total ⑧ frames). So, some of these frames collide because multiple frames are in Contention for shared channel. The frame 1.1 from S1 (ie station 1) & frame 3.2 from S3 survive & remaining collide. We need to mention that Even if 1-bit of a frame Co-Exists on channel with one bit from another frame, there is a Collision & both will be destroyed.

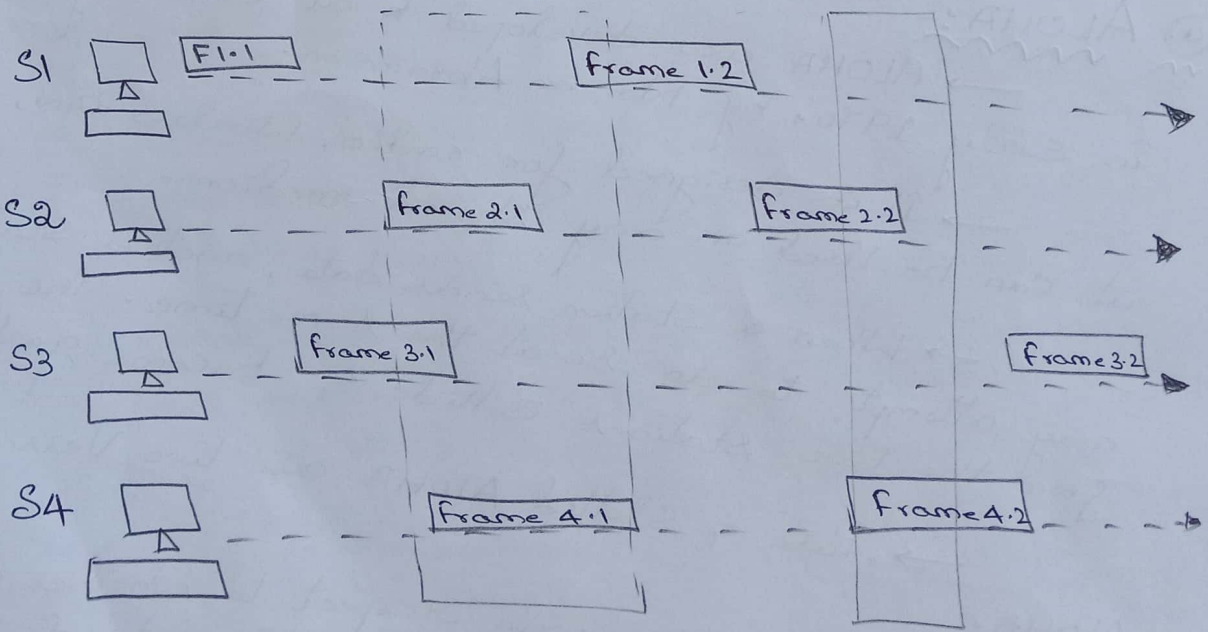


fig ⓐ: frames in Pure ALOHA network.

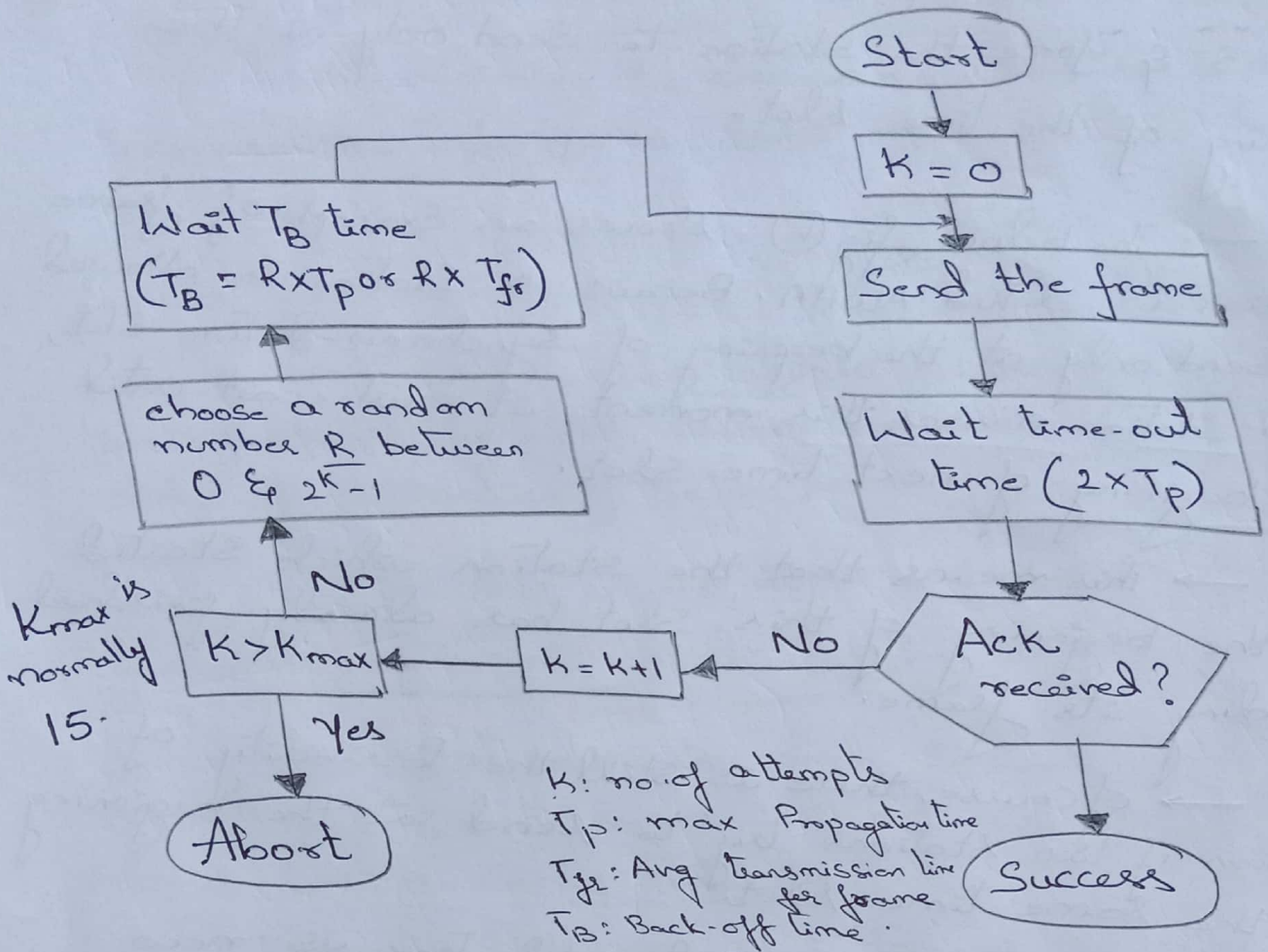


fig ⑥ Procedure for Pure ALOHA Protocol.

2. Slotted ALOHA:-

→ Pure ALOHA has a Vulnerable time of $2 \times T_{fr}$. 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 1972, Robert's Proposed a system called Slotted ALOHA.

→ In Slotted ALOHA, we divide the time into slots of T_{fr} s & force the station to send only at the beginning of the time slot.

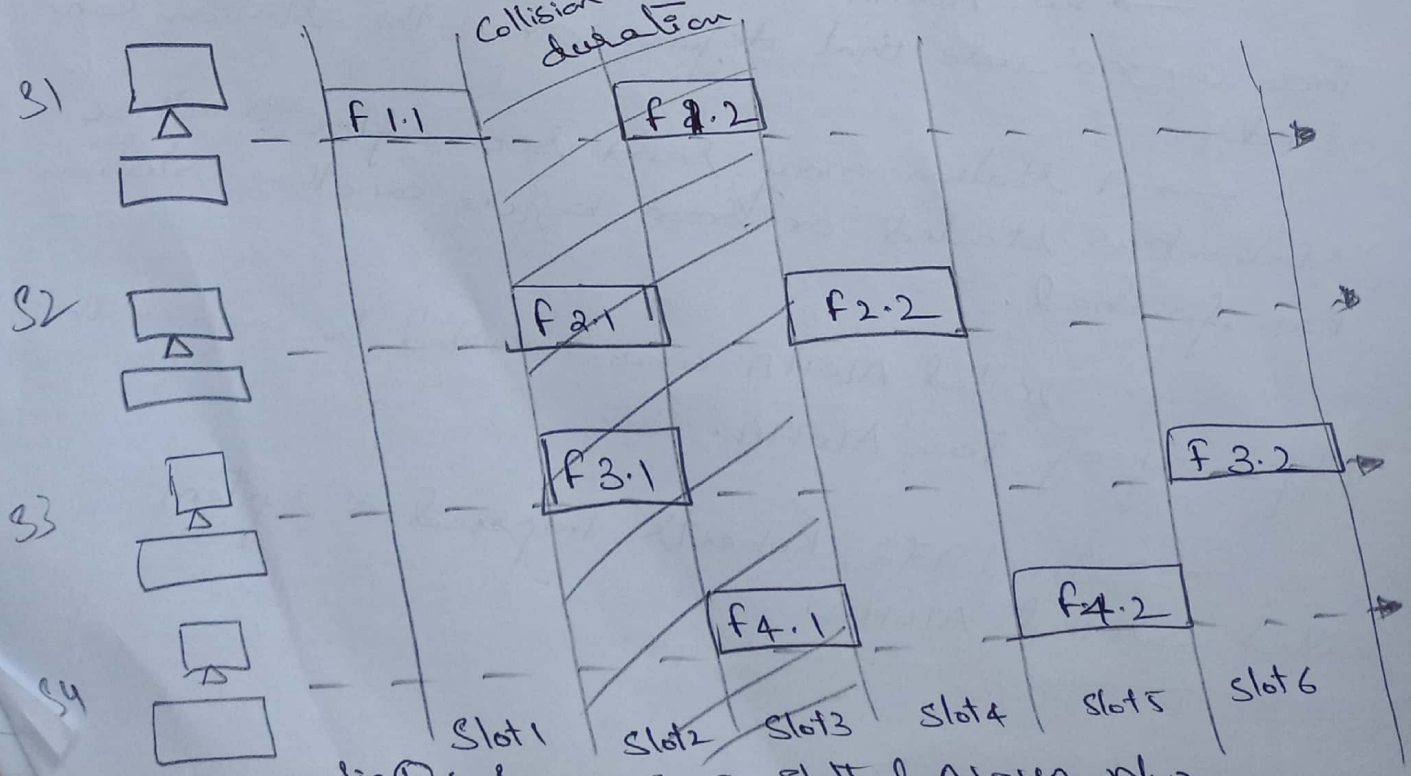
→ The below fig ⑥, shows an Example of frame Collision in Slotted ALOHA. Because a Station is allowed to send only at the begining of Synchronized time slot, if a station misses this moment, it must wait until the begining of next time slot.

→ This means that the station which started at the beginning of this slot has already finished Sending its frame.

→ Of course, there is still the Possibility of Collision if two stations try to send at the beginning of the same time slot.

→ However, the Vulnerable time is now reduced to <u>One-half</u>, Equal to T_{fr}.

Slotted ALOHA Vulnerable Time = T_{fr}



<u>fig ⑥: frames in a Slotted ALOHA n/w.</u>

(b) CSMA :-

↳ To minimize the chance of collision & therefore increase the Performance, CSMA method was developed.

→ The chance of collision reduces, if station Senses the medium before trying to use it.

→ Carrier Sense Multiple Access (CSMA) requires that each station first listen to medium (or check the state of the medium) before sending.

→ In other words, CSMA is based on the Principle "Sense before transmit" or "listen before talk".

→ Protocols in which stations listen for a carrier (i.e a transmission) & act accordingly are called Carrier Sense Protocols.

→ ~~Kel~~ Kleinrock & Tobagi in 1975 have analyzed Several such Protocols.

• Persistant CSMA :-

→ When a station has data to Send, it first listens to channel to see if anyone Else is transmitting at that moment.

→ If channel is busy, the station waits until it becomes idle.

→ When station detects an idle channel, it transmits a frame.

→ If collision occurs, the station waits a random amount of time & starts all over again. The Protocol is called 1- Persistant.

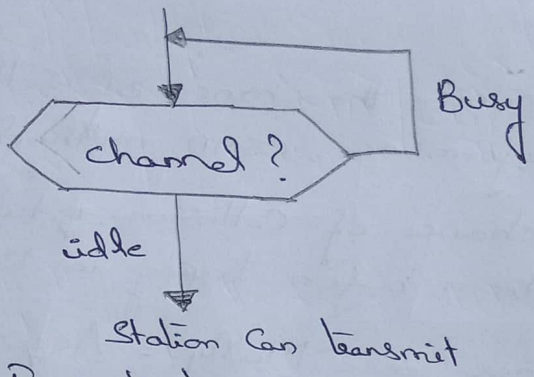


Fig: 1-Persistent.

• Non-Persistent :-

→ When Station has a frame to send, first it senses the channel, if it is idle, it sends immediately.

→ If channel is busy, it waits for random amount of time & then senses the channel again.

→ In this approach it reduces the chance of collision because it is unlikely that two or more Stations will wait the same amount of time & retry to send simultaneously.

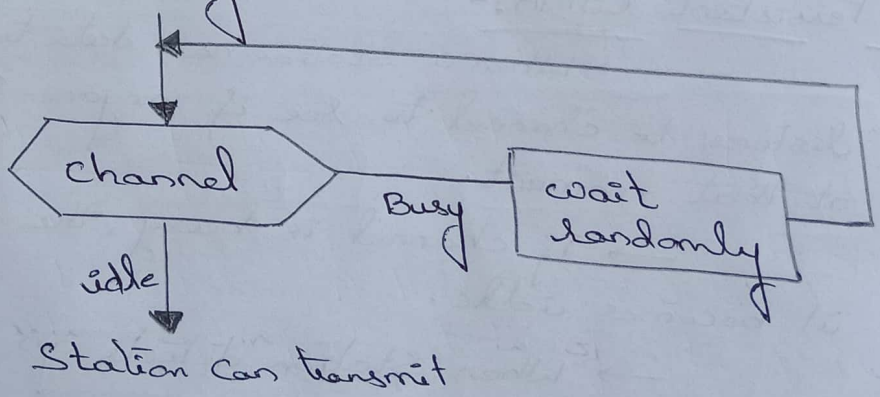


fig (b): non-Persistant

fig (b): frames in a slotted ALOHA n/w.

• P- Persistent:-

→ It is used if the channel has time slots with a slot duration Equal to or greater than the maxim Propogation time.

→ This approach combines the advantages of other two Strategies.

→ It reduces the chance of Collision & improve Efficiency.

→ In this method, after the station finds the line idle it follows these steps:

1. with Probability P, the station sends its frame
2. With Probability $q = 1-P$, the station waits for the beginning the next time slot & checks the channel again.
   - (a) If the channel is idle, go to step (1)
   - (b) if the channel is busy, it acts as though a Collision has occured & uses the backoff Procedure.

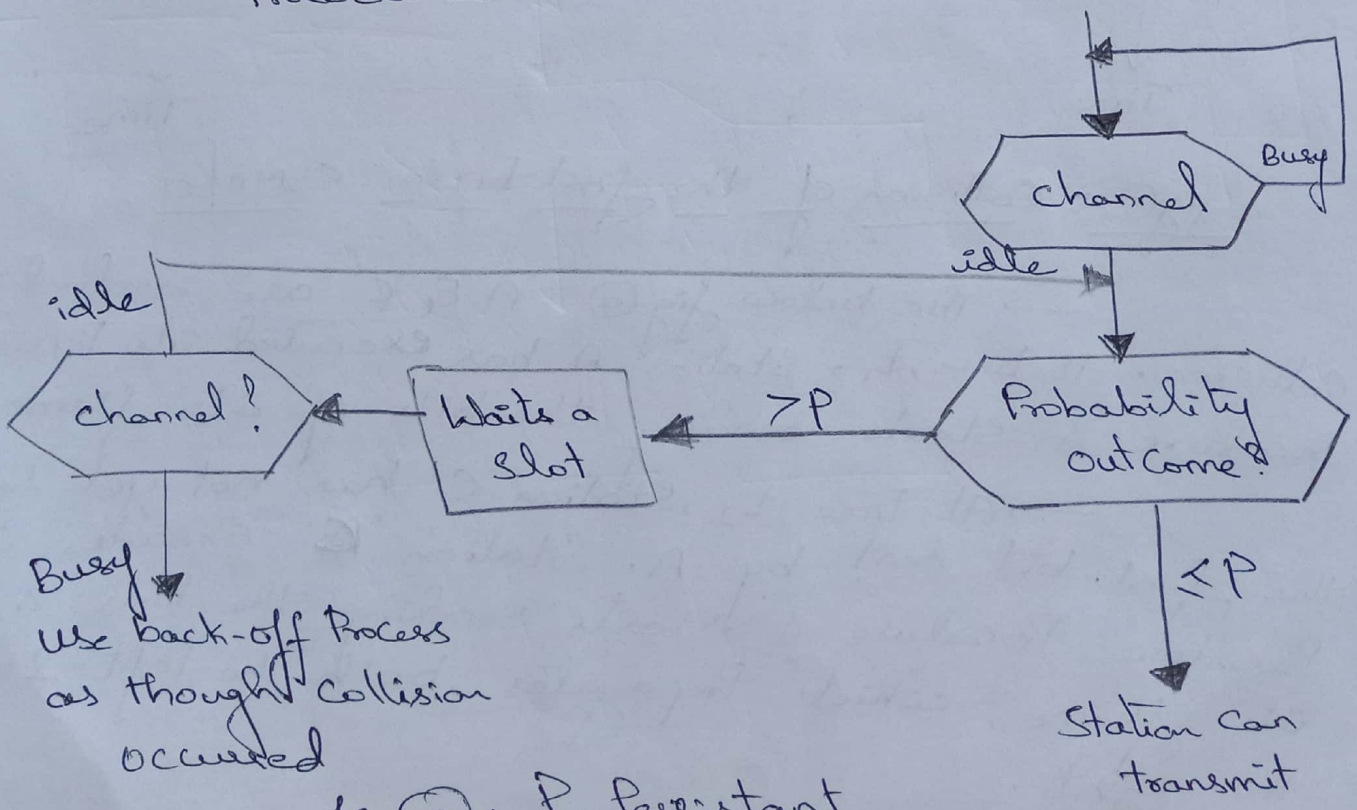
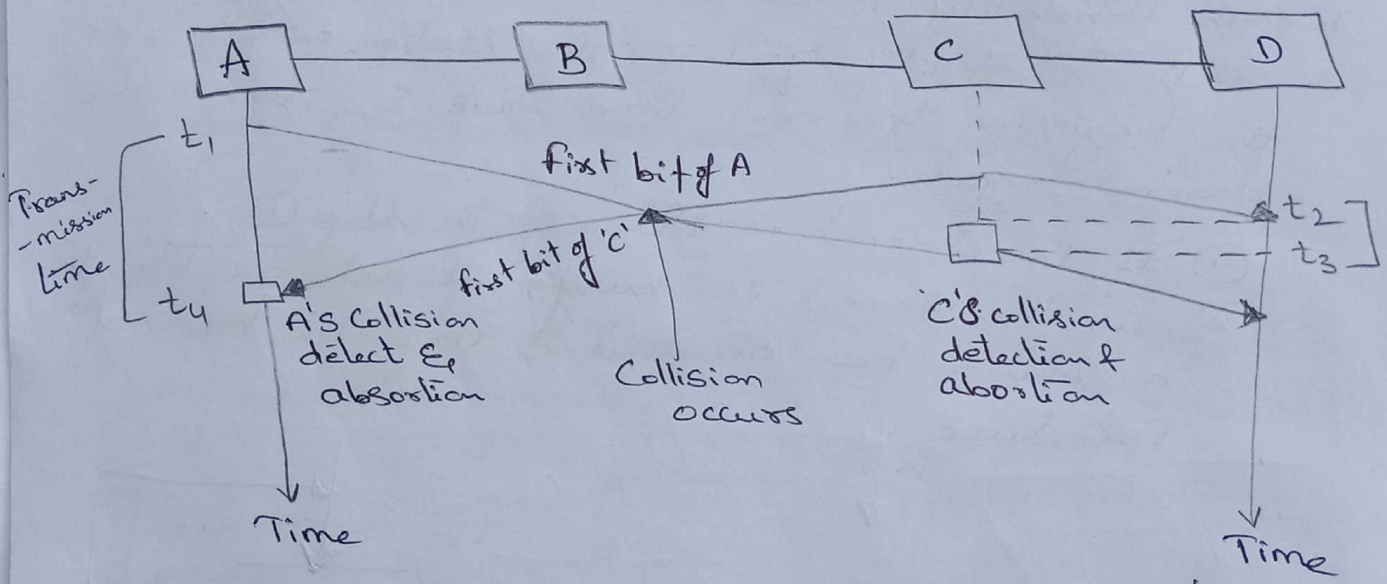


fig (c): P- Persistant

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

⟶ 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 so, the station is finished. If however, there is a collision, the frame is sent again.



fig@: Collision of the first bit in CSMA/CD.

⟶ The below fig@ 'A' & 'C' are involved in collision. At time t_1, station A has executed its Persistence Procedure & starts sending the bits of its frame.

⟶ At time t_2, station 'C' has not yet sensed the first bit sent by A. Station 'C' executes its Persistence Procedure & starts sending the bits in its frame, which propagates both to the left & to the right.

fig⑥: frames in a slotted ALOHA n/w.

→ The collision occurs a sometimes after time t_2. Station 'C' detects the collision at time t_3 when it receives the first bit of A's frame.

→ Station 'C' immediately (or after some time) aborts transmission.

→ Station 'A' detects the collision at time t_4 when it receives the first bit of C's frame, it also immediately aborts transmission.

→ by above fig(a), Station 'A' transmits for the duration $t_4 - t_1$, 'C' transmits for the duration $t_3 - t_2$.

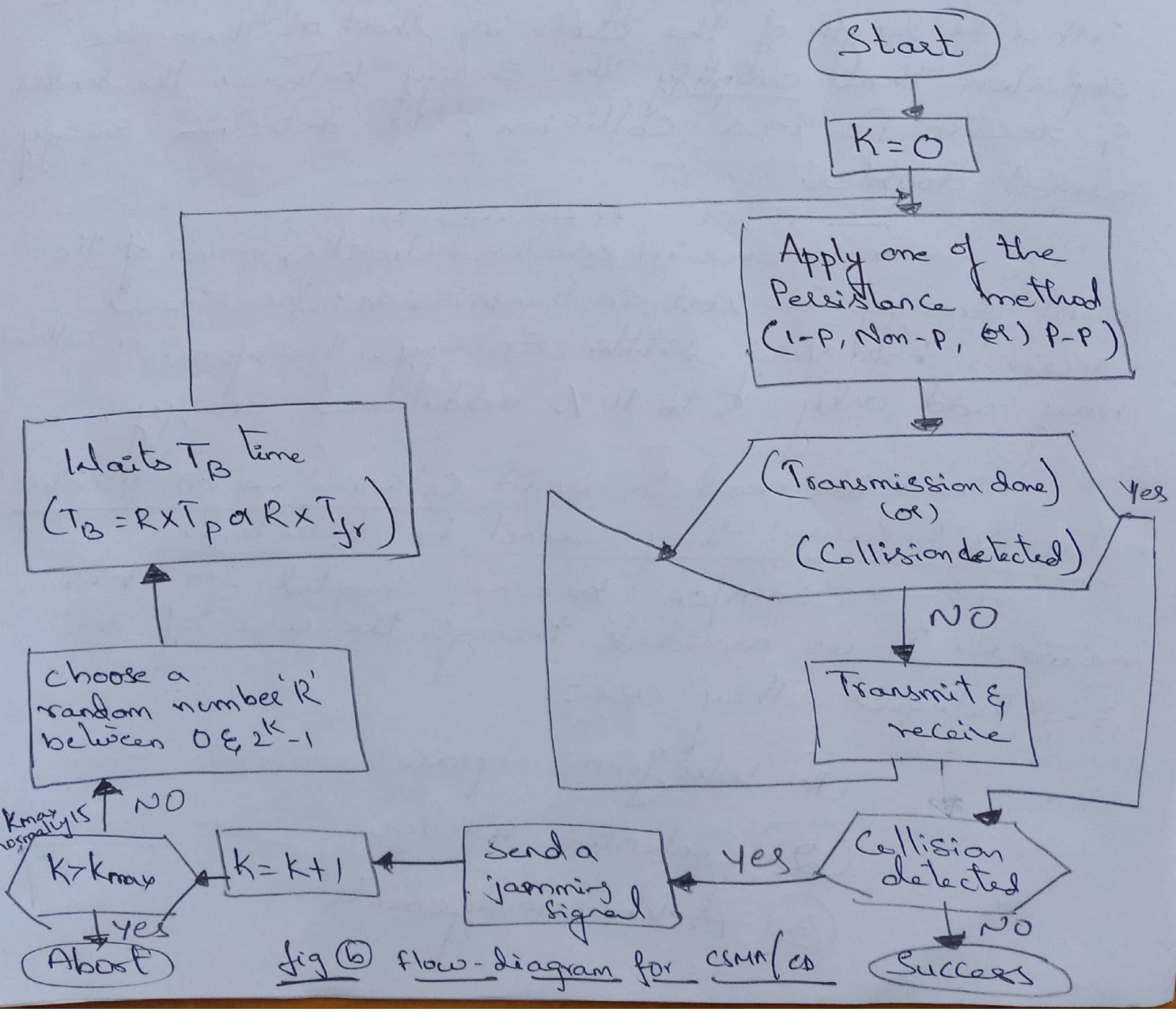


fig (b) flow-diagram for CSMA/CD

(d) Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA):-

→ The basic idea behind CSMA/CA is that a station needs to be able to receive while transmitting to detection a collision.

→ When there is a collision, station receives only one signal (i.e its own signal), Simultaneously if no collision it receives two signals < own signal / signal to be transmitted.

→ In wired networks, the received signal has almost same Energy as the sent signal because either the length of the cable is short or there are repeaters that amplify the energy between the sender & receiver i.e in a collision, the detected energy almost doubles.

→ However in wireless networks, much of the sent energy is lost in transmission. The signal received has very little Energy. Therefore, a collision may add only 5 to 10% additional Energy.

→ We need to avoid collision ~~in~~ on wireless networks because they cannot be detected.

→ CSMA/CA ~~he~~ was invented for this network. It is avoided through the use of 3 - Strategies. They are:

1. Inter frame Space
2. Contention Window
3. Acknowledgements

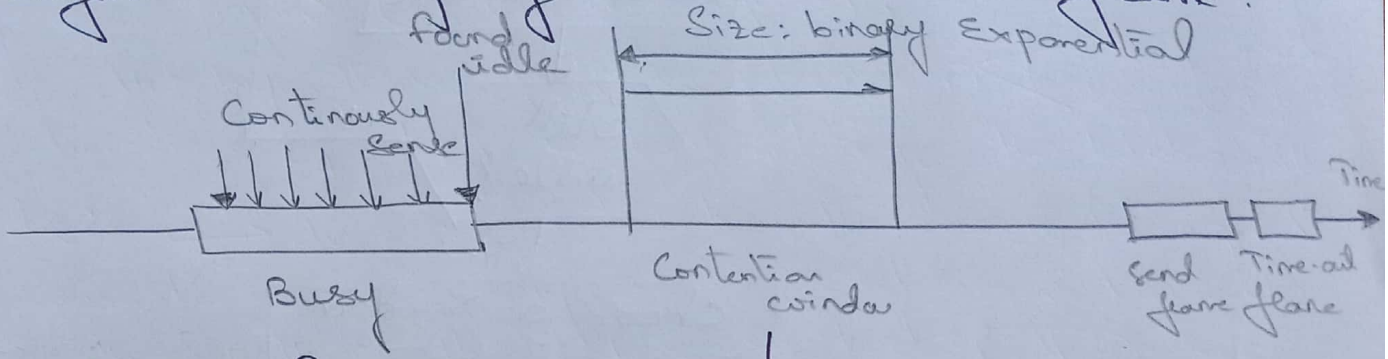
(b) frames in a slotted ALOHA [illegible]

1. Interframe Space (IFS):-

→ Collision are avoided by deferring transmission Even if the channel is found idle.

→ It waits for a Period of time called Interframe Space or IFS.

→ In CSMA/CA, the IFS can also be used to define the Priority of the station or a frame.



fig(a): Timing in CSMA/CA

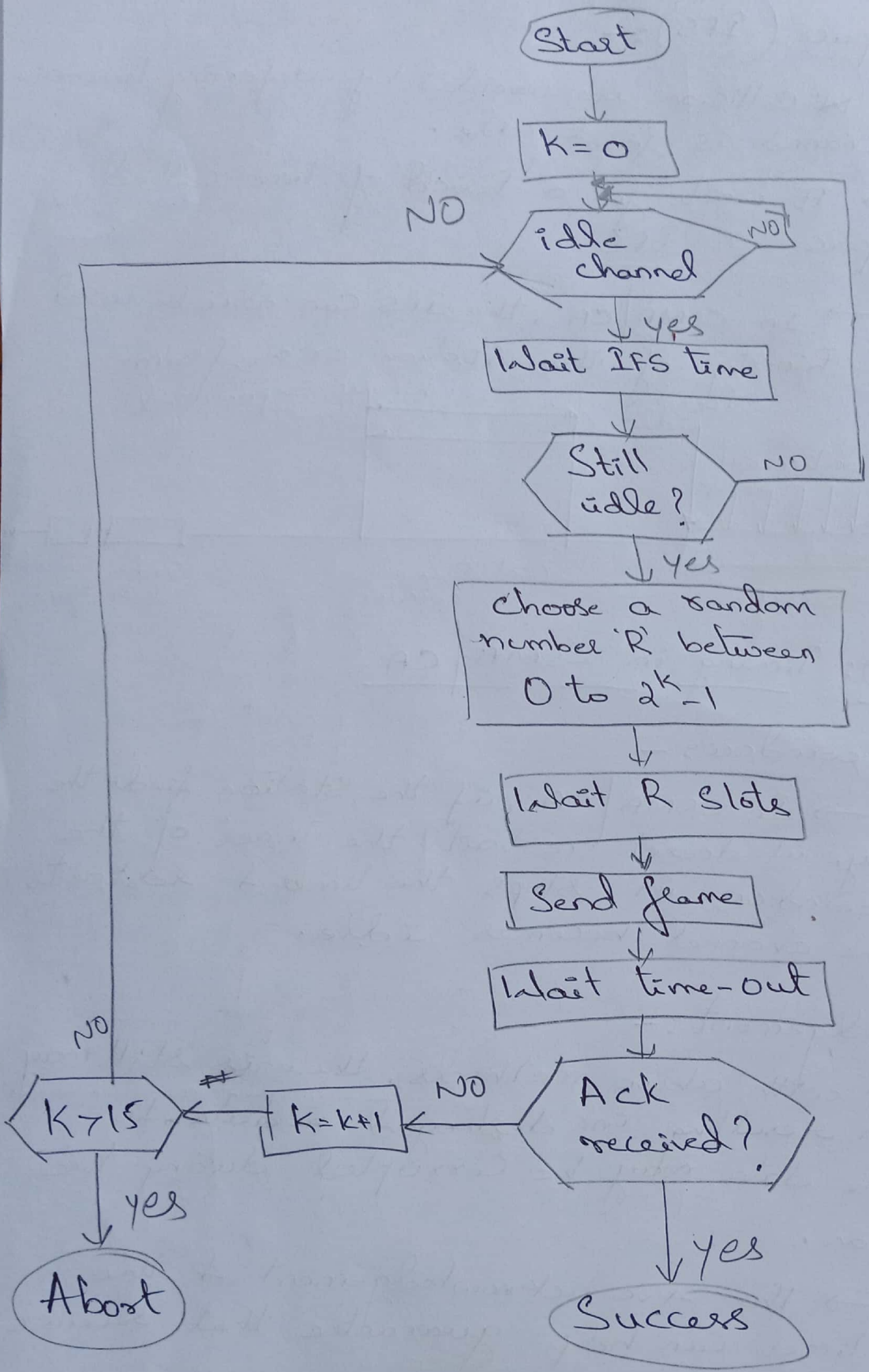
2. Contention window:-

→ In CSMA/CA, if the station finds the channel busy, it doesnt restart the timer of the Contention window, it stops the timer & restarts it when the channel becomes idle.

(3) Acknowledgement:-

→ With above methods, there is still may be collision resulting in destroyed data. In addition, the data may be corrupted during the transmission.

→ The +ve acknowledgement & the time-out timer can help guarantee that receiver has received the frame.



fig@:- flow-diagram for CSMA/CD