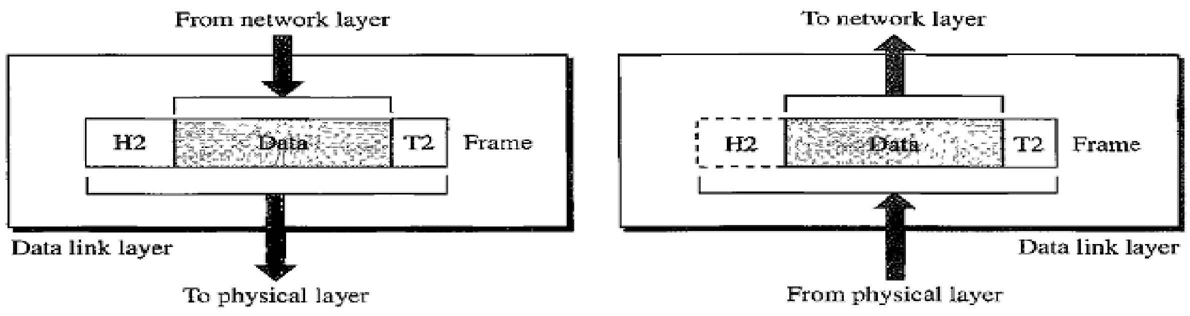
Unit-II

Data Link Layer

The responsibility of the **Physical Layer** is to transmit the unstructured raw bit stream over a physical medium.

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.

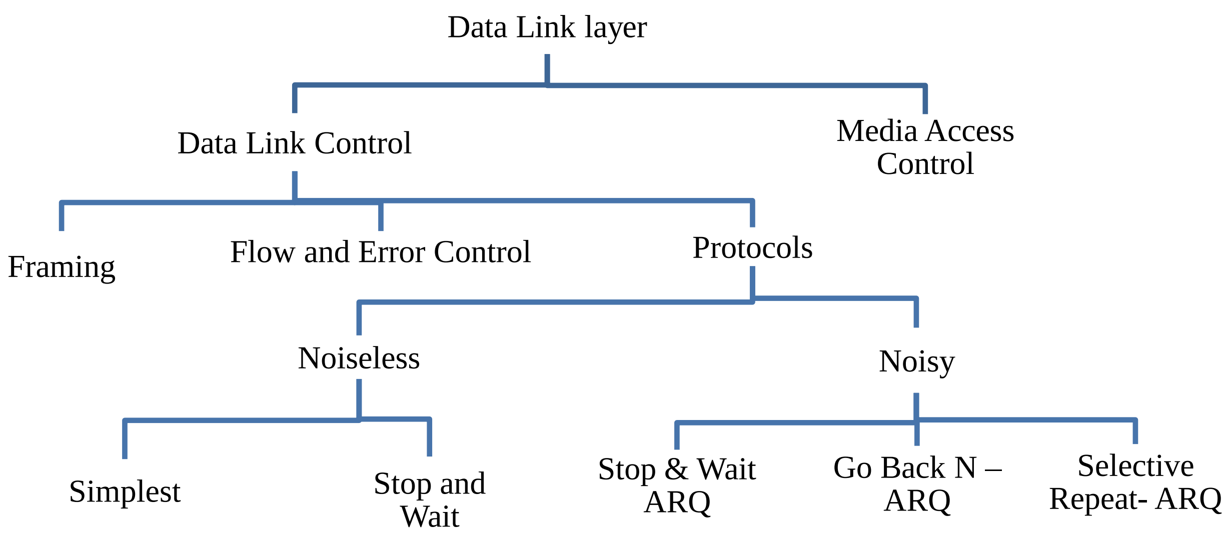
Flow Control
If the rate at which the data are absorbed by the receiver is less than the rate at which data are produced in the sender, the data link layer imposes a flow control mechanism to avoid overwhelming the receiver.

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:



Note:

1. Physical layer transfers bits in the form of a signal from the source to the destination.
2. The data link layer converts bits into frames, so that each frame is distinguishable from another.

Framing

- Framing is a process of adding source address and destination address to message.
- The destination address defines where the packet is to go.
- The sender or source address helps the receiver to acknowledge the receipt.
- If the message is large we will divide the message into several frames. Because larger frames causes flow and error control problems, if any single bit error occurs we have to retransmit the entire message this consumes a lot time.
- If a message is divided into smaller frames the single bit error can effect only one frame.

Framing can be done in 2 ways:

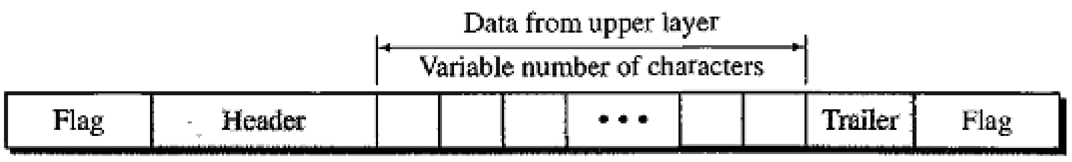
- **Fixed size framing:** The size of the frame is fixed for all the frames. There is no need to define the boundaries of a frame.
- **Variable size framing:** In variable-size framing, we need a way to define the end of the frame and the beginning of the next frame.

There are 2 approaches are used for variable size framing:

1. Character Stuffing (A Character-Oriented Approach)
2. Bit Stuffing (A Bit-Oriented Approach)

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.

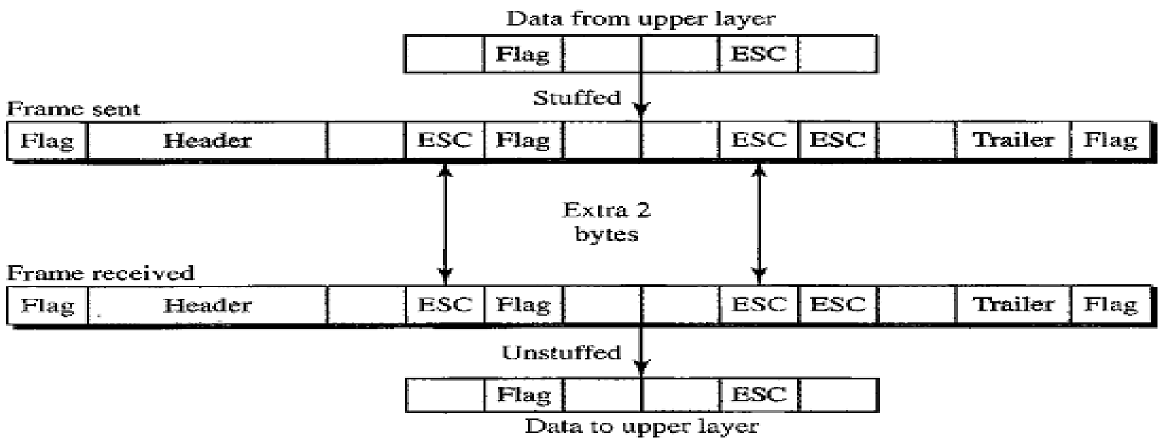
Disadvantages of Character Stuffing

- 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

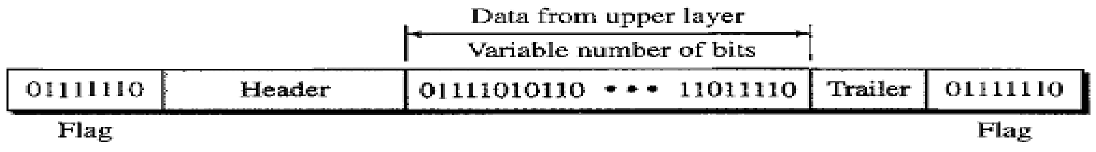
- The universal coding systems (Unicode) in use today have 16-bit and 32-bit characters that conflict with 8-bit characters.
- Character stuffing deals with 8-bit characters but todays systems using 16 bits, 32 bits and 64 bit characters hence there will be conflict.

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

Bit stuffing

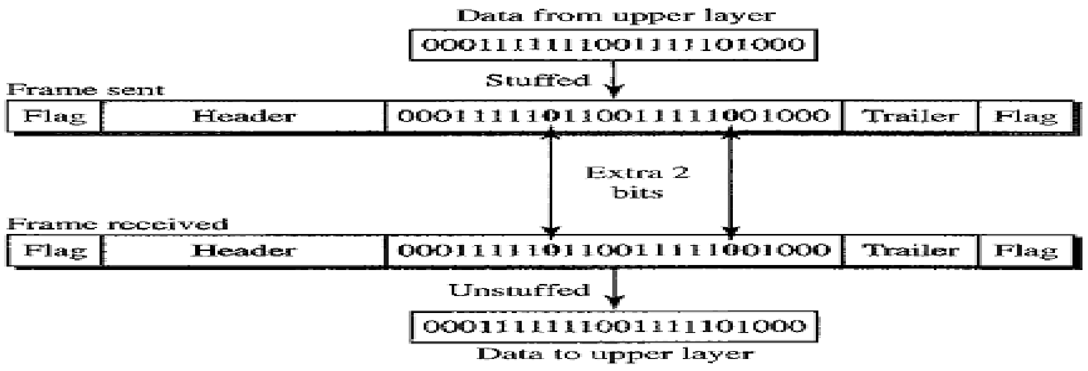
- It is used for text, graphic, audio, video, and so on.
- In bit stuffing the data section of a frame is a sequence of bits to be interpreted by the upper layer.
- In addition to header and trailer, we need a delimiter to separate one frame from other frame.
- Most protocols use a special 8-bit pattern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

Error Detection and Correction

Data can be **Corrupted** during Transmission. Applications require that errors to be **Detected** and **Corrected**. To detect or correct errors **Redundancy** concept is used.

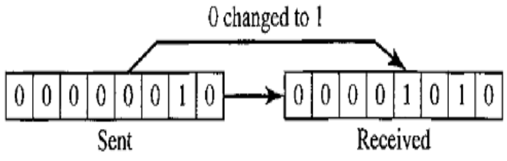
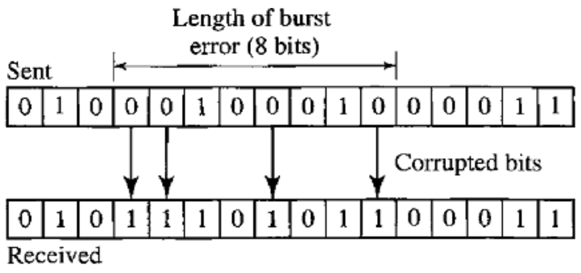
That means:

- Networks must be able to transfer data from one device to another with acceptable accuracy.
- For most applications, a system must guarantee that the data received are identical to the data transmitted.
- Any time data are transmitted from one node to the next node, they can become corrupted in passage. Many factors can alter one or more bits of a message.
- Hence applications require a mechanism for detecting and correcting errors.

Note: Some applications can tolerate a small level of **Error**.
For Example: Random errors in audio or video transmissions may be tolerable, but when we transfer text, we expect a very high level of accuracy.

Types of Errors

Whenever bits flow from one point to another, they are subject to unpredictable changes because of interference. (Noise in the interface makes errors)

Single Bit Error	Burst Error
Only 1 bit of a given data unit (such as a byte, character, or packet) is changed from 1 to 0 or from 0 to 1. 	2 or more bits in the data unit have changed from 1 to 0 or from 0 to 1.  Length of burst error (8 bits) Sent Corrupted bits Received

- A burst error is more likely to occur than a single-bit error.
- The duration of noise is normally longer than the duration of 1 bit, which means that when noise affects data, it affects a set of bits.
- The number of bits affected depends on the data rate and duration of noise.
- For example, if we are sending data at I kbps, a noise of 11100 s can affect 10 bits; if we are sending data at 1Mbps, the same noise can affect 10,000 bits.

Redundancy

- To be able to detect or correct errors, we need to send some extra bits (Redundant bits) with our data.
- These redundant bits are added by the sender and removed by the receiver.
- Redundant bits allow the receiver to detect or correct corrupted bits.

Detection versus Correction

The correction of errors is more difficult than the detection.

Error Detection	Error Correction
1. In error detection, it is used to find any errors are occurred.	1. In error correction, we need to know the exact number of bits that are corrupted and their location in the message.
2. We are not interested in the number of errors.	2. The number of the errors and the size of the message are important factors.
3. A single-bit error is the same as a burst error.	3. If we need to correct one single error in an 8-bit data unit, we need to consider eight possible error locations; if we need to correct two errors in a data unit of the same size, we need to consider 28 possibilities.

Forward Error Correction Versus Retransmission

There are two main methods of error correction:

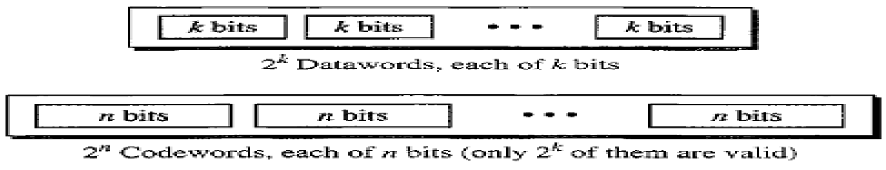
Forward Error Correction	**Retransmission**
1. Forward error correction is the process in which the receiver tries to guess the message by using redundant bits. 2. This is possible only when the number of errors is small.	1. Retransmission is a technique in which the receiver detects the occurrence of an error and asks the sender to resend the message. 2. Resending is repeated until a message arrives that the receiver believes is error-free

Coding Versus Decoding

Coding	**Decoding**
The sender adds redundant bits through a process that creates a relationship between the redundant bits and the actual data bits.	The receiver checks the relationships between the two sets of bits to detect or correct the errors.

BLOCK CODING

- In block coding, we divide our message into **Blocks**, each of ***k*** **bits**, called **Datawords**.
- We add ***r*** **redundant bits** to each block to make the length ***n = k + r***. The resulting *n-bit* blocks are called **Codewords**.
- With ***k*** **bits**, we can create a combination of 2^k datawords.
- With ***n*** **bits**, we can create a combination of 2^n codewords.
- Since ***n*** > ***k***, the number of possible codewords is larger than the number of possible datawords.
- The block coding process is one-to-one; the same dataword is always encoded as the same codeword.
- This means that we have $2^n - 2^k$ codewords that are not used. These unused codewords are called invalid or illegal codewords.


k bits
k bits
• • •
k bits
2^k Datawords, each of k bits
n bits
n bits
• • •
n bits
2^n Codewords, each of n bits (only 2^k of them are valid)

Example:

In a coding scheme, ***k*** **=4** and ***n*** **=5**. As we saw, we have 2^k **= 16** datawords and 2^n **=32** codewords. We saw that 16 out of 32 codewords are used for message transfer and the rest are either used for other purposes or unused.

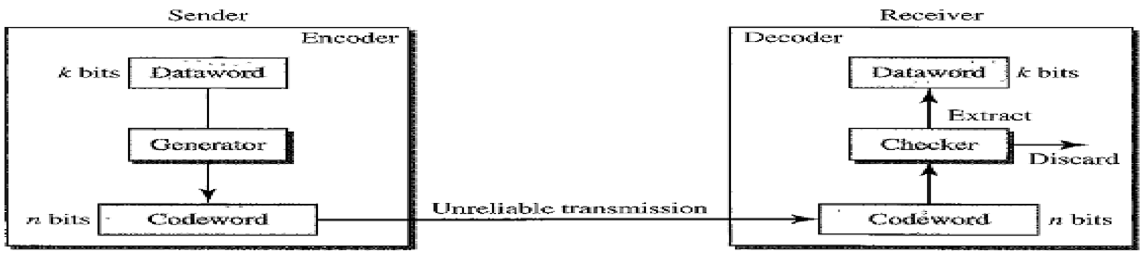
Error Detection

In Block coding the errors can be detected by meeting the following two conditions:

In Error Detection the receiver can detect a change in the original word.

1. The receiver has (or can find) a list of valid codewords.
2. The original codeword has changed to an invalid one.

- The sender creates codewords out of datawords by using a generator that applies the rules and procedures of encoding.
- Each codeword sent to the receiver may change during transmission. If the received codeword is the same as one of the valid codewords, the word is accepted; the corresponding dataword is extracted for use.
- If the received codeword is not valid, it is discarded.

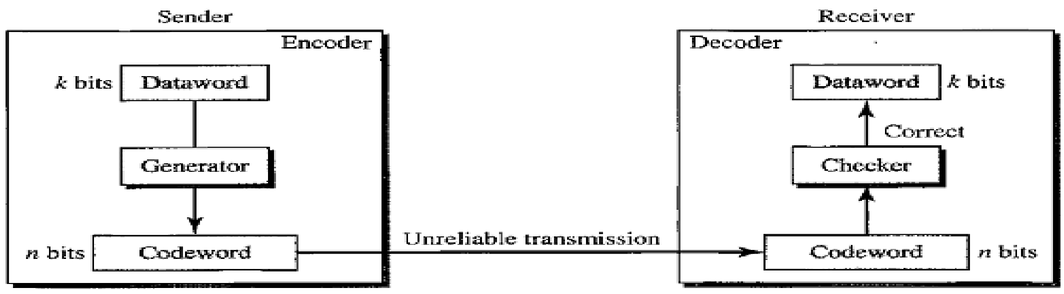


Note: If the codeword is corrupted during transmission but the received word still matches a valid codeword, the error remains undetected. This type of coding can detect only single errors. Two or more errors may remain undetected.

Error Correction

In error detection, the receiver needs to know only that the received codeword is invalid. I

n error correction the receiver needs to find (or) guess the original codeword sent.



<u>Hamming Distance (or) Hamming Codes</u>

- Hamming distance is central concepts in coding for error control.
- The Hamming distance between two words of the same size is the number of differences between the corresponding bits.
- Hamming distance between two words *x* and *y* is represented as *d(x, y).*
- The Hamming distance can easily be found if we apply the XOR operation on the two words and count the number of 1's in the result.

Note: Hamming distance is a value greater than zero.

Example:

Find the Hamming distance between two pairs of words

1. The Hamming distance d(000, 011) is 2 because 000 $\oplus$011 is 011 (two 1's)
2. The Hamming distance d(10101, 11110) is 3 because 10101 $\oplus$11110 is 01011 (three 1's).

Minimum Hamming Distance (d_{min})

The Minimum Hamming distance is the smallest Hamming distance between all possible pairs in a set of words.

Example : Find the minimum Hamming distance of the coding scheme.

d(00000, 01011) = 3	d(01011, 10101) = 4	d(00000,10101) = 3
d(0l011, 11110) = 3	d(00000, 11110) = 4	d(10101, 11110) =3

The d_{min} in this case is 3.

Note: Any coding scheme needs to have at least three parameters:

1. The codeword size *n*
2. The dataword size *k*
3. The minimum Hamming distance d_{min}.

A coding scheme C is written as **C(n, k)** with a separate expression for d_{min}

For example: Let Coding scheme *C(3, 2)* with d_{min} = 2 and

Let coding scheme *C(5,* 2) with d_{min}= 3.

To guarantee the detection of up to **p** errors in all cases, the minimum Hamming distance in a block code must be d_{min} = p+1.

<u>LINEAR BLOCK CODES</u> (LBC)

A Linear Block Code is a code in which the exclusive OR (addition modulo-2) of two valid codewords creates another valid codeword.

The minimum Hamming distance is the number of 1's in the nonzero valid codeword with the smallest number of 1's.

<u>Types of LBC</u>

1. Parity Check Code
2. Cyclic Redundancy Code (CRC)

<u>Parity-Check Code</u>

- In this code, a **k-bit** dataword is changed to an **n-bit** codeword where ***n* = *k* + 1.**
- The extra bit is called the parity bit is selected to make the total number of 1's in the codeword even.
- The minimum Hamming distance for this category is $d_{min} = 2$, which means that the code is a single-bit error-detecting code; it cannot correct any error.

<u>Sender Side</u>

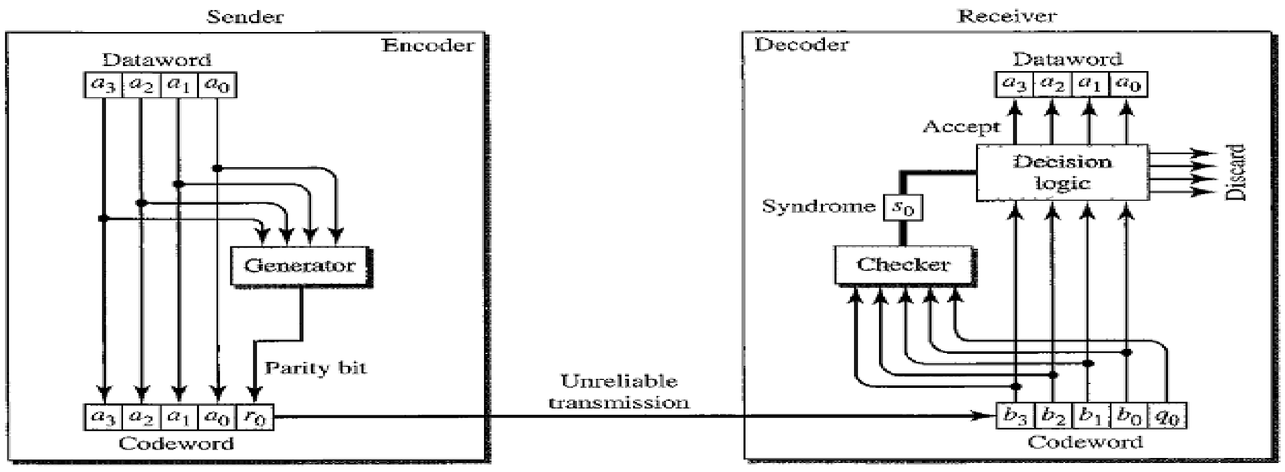
- The encoder uses a generator that takes a copy of a 4-bit dataword *(*a_0, a_1 a_2 and a_3*)* and generates a parity bit r_0The dataword bits and the parity bit create the 5-bit codeword.
- The parity bit that is added makes the number of 1's in the codeword even.
- This is normally done by adding the 4 bits of the dataword and performing modulo-2 operation on the result.

$$\boxed{r_0=a_3+ a_2+ a_1+ a_0 \text{ (modulo 2)}}$$

- If the number of 1s is even, the result is 0; if the number of 1s is odd, the result is 1. In both cases, the total number of 1s in the codeword is even.

Note: The sender sends the codeword which may be corrupted during transmission.

Below figure shows a possible structure of an encoder (at the sender) and a decoder (at the receiver).



Receiver Side

- The receiver receives a 5-bit word. The checker at the receiver does the same thing as the generator in the sender with one exception: The addition is done over all 5 bits. The result is called the syndrome and it is just 1 bit.
- The syndrome is 0 when the number of 1's in received codeword is even.
- The syndrome is 1 when the number of 1's in received codeword is odd.

$$\boxed{s_0 = b_3 + b_2 + b_1 + b_0 + q_0 \text{ (modulo 2)}}$$

- The syndrome is passed to the decision logic analyzer. If the syndrome is 0, there is no error in the received codeword; the data portion of the received codeword is accepted as the dataword;
- If the syndrome is 1, the data portion of the received codeword is discarded. The dataword is not created.

Example: Consider the code in below Table is a parity-check code with $k = 4$ and $n = 5$. [i.e. C(n,k) = C(5,4)]

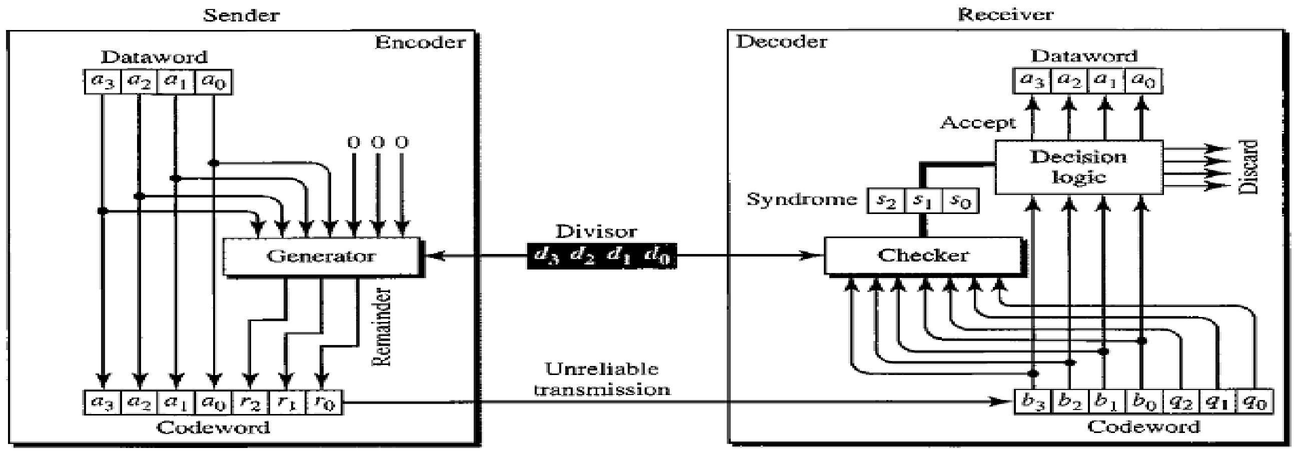
Datawords	*Codewords*	*Datawords*	*Codewords*
0000	00000	1000	10001
0001	00011	1001	10010
0010	00101	1010	10100
0011	00110	1011	10111
0100	01001	1100	11000
0101	01010	1101	11011
0110	01100	1110	11101
0111	01111	1111	11110

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_2r_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).

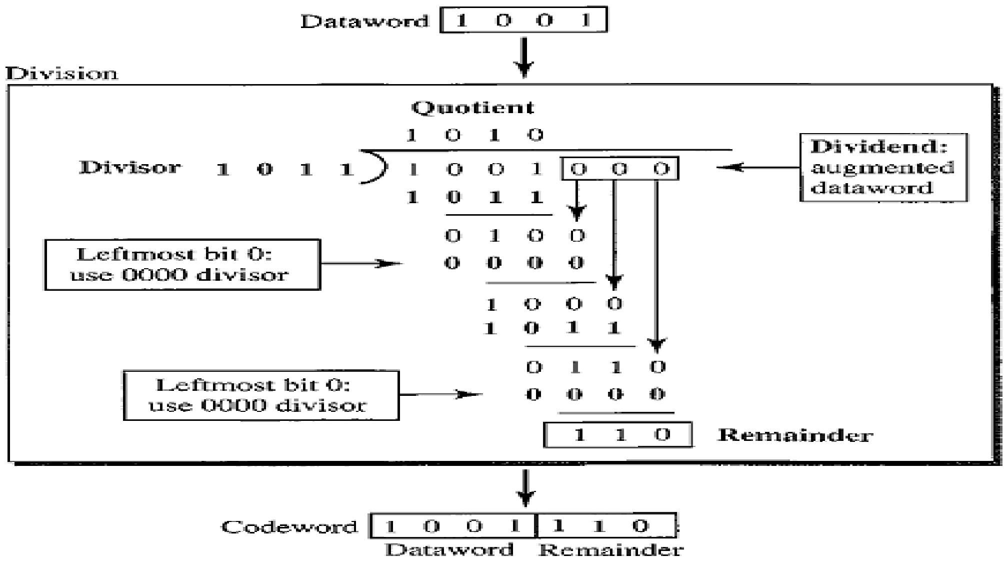
Example: A CRC code with C(7, 4)

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.

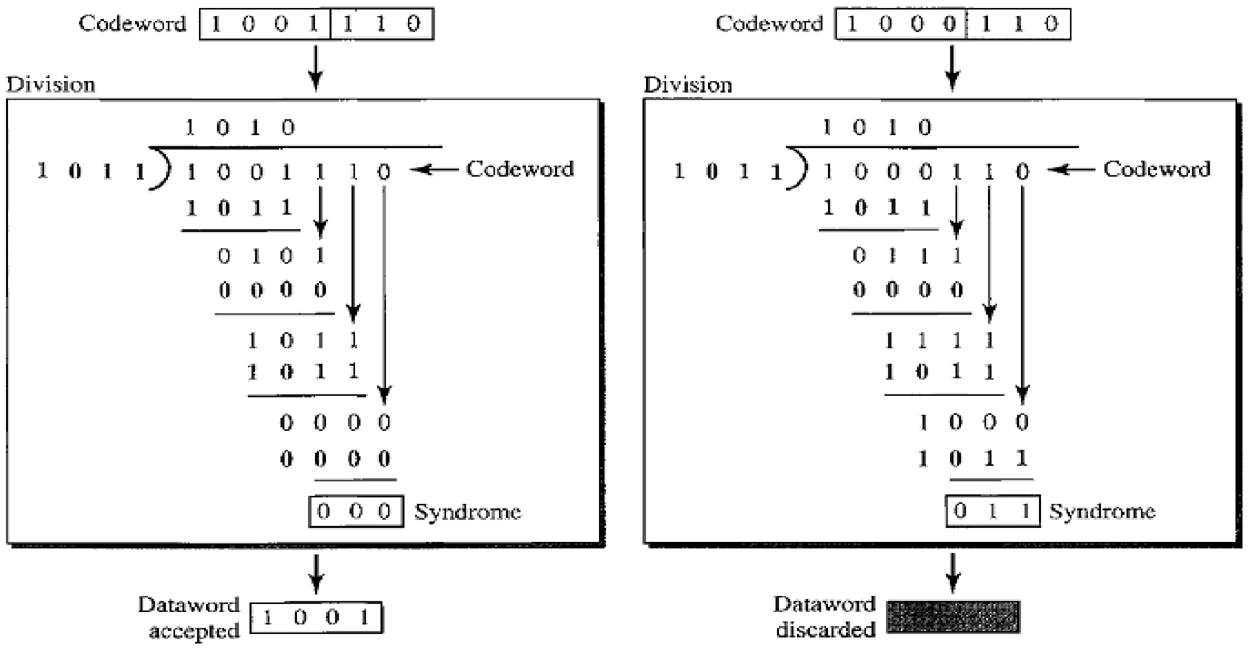
Decoder

- 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).



FLOW CONTROL

- Flow control is a set of procedures that tells the sender how much data it can transmit before it must wait for an acknowledgment from the receiver.
- The flow of data must not be allowed to overwhelm the receiver.
- Any receiving device has a limited speed at which it can process incoming data and a limited amount of memory in which to store incoming data.
- The receiving device must be able to inform the sending device before those limits are reached and to request that the transmitting device send fewer frames or stop temporarily.
- Incoming data must be checked and processed before they can be used.
- The rate of receiving data is slower than the rate of transmission of data.
- For this reason, each receiving device has a block of memory, called a *buffer,* reserved for storing incoming data until they are processed.
- If the buffer begins to fill up, the receiver must be able to tell the sender to halt transmission until it is once again able to receive.

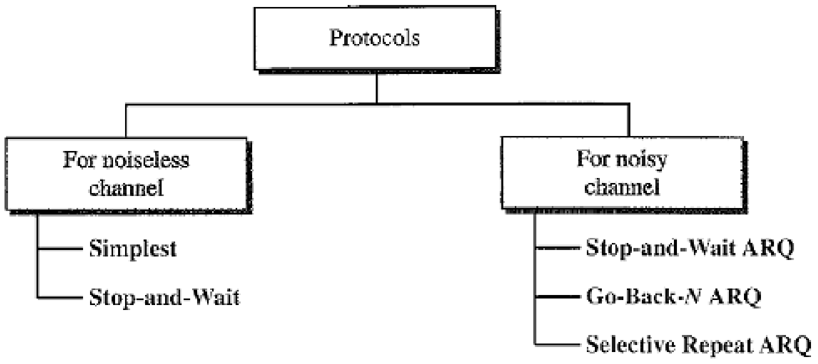
ERROR CONTROL

- Error control is both error detection and error correction.
- It allows the receiver to inform the sender of any frames lost or damaged in transmission and coordinates the **Retransmission** of those frames by the sender.

- At any time an error is detected in an exchange, specified frames are retransmitted. This process is called **Automatic Repeat Request** (**ARQ**).

PROTOCOLS

The protocols are normally implemented in software by using the common programming languages.



Noiseless Channels

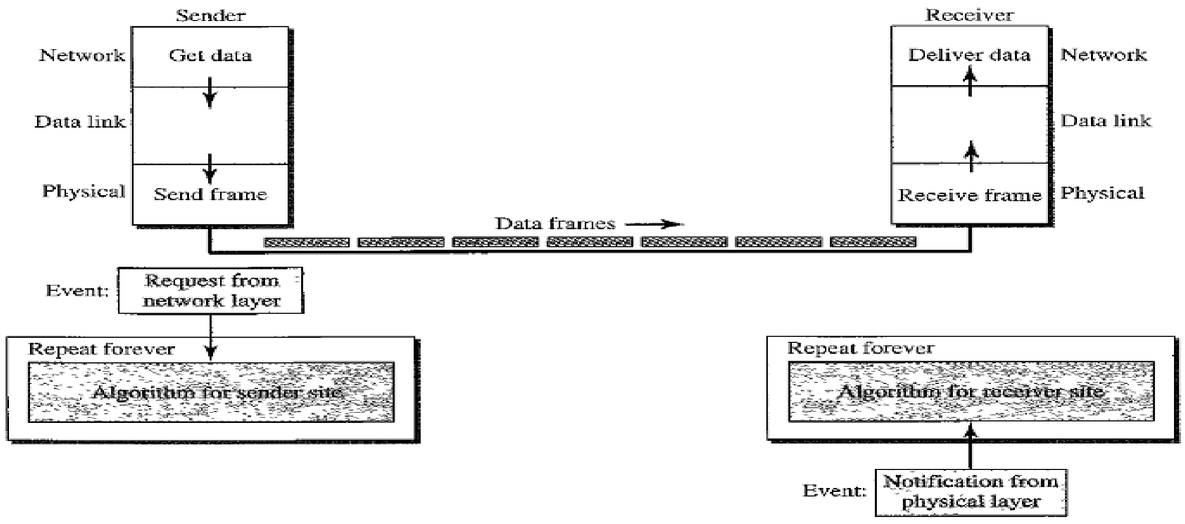
Noiseless Channel is an ideal channel in which no frames are lost, duplicated, or corrupted. There are two protocols for this type of channel:

1. Simplest Protocol
2. Stop and Wait Protocol

Note: In real world there are no Noiseless channels.

Simplest Protocol

- Simplest Protocol has no flow control and no error control mechanism.
- It is a unidirectional protocol in which data frames are traveling in only one direction-from the sender to receiver.
- It is applicable when the receiver can never be overwhelmed with incoming frames that are sent by the sender.
- The receiver can receives any frame with a very short span of processing time.
- The data link layer of the receiver immediately removes the header from the frame and delivers the data packet to its network layer.



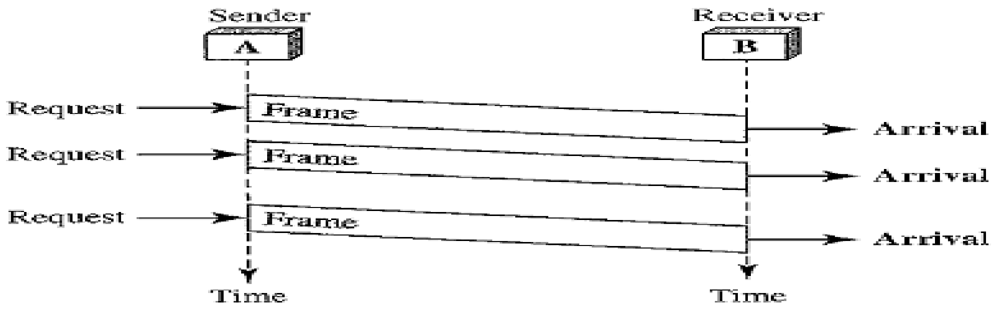
Design

- The data link layer at the **Sender site** gets data from its network layer, makes a frame out of the data, and sends it.
- The data link layer at the **Receiver site** receives a frame from its physical layer, extracts data from the frame, and delivers the data to its network layer.
- The data link layers of the sender and receiver provide transmission services for their network layers.
- The data link layers use the services provided by their physical layers (such as signaling, multiplexing, and so on) for the physical transmission of bits.
- There is no need for flow control in Simplest Protocol scheme.

Simplest protocols are implemented as an Event driven Procedure.

- The procedure at the sender site is constantly running; there is no action until there is a request from the network layer.
- The procedure at the receiver site is also constantly running, but there is no action until notification from the physical layer arrives.
- Both procedures are constantly running because they do not know when the corresponding events will occur.

Flow diagram



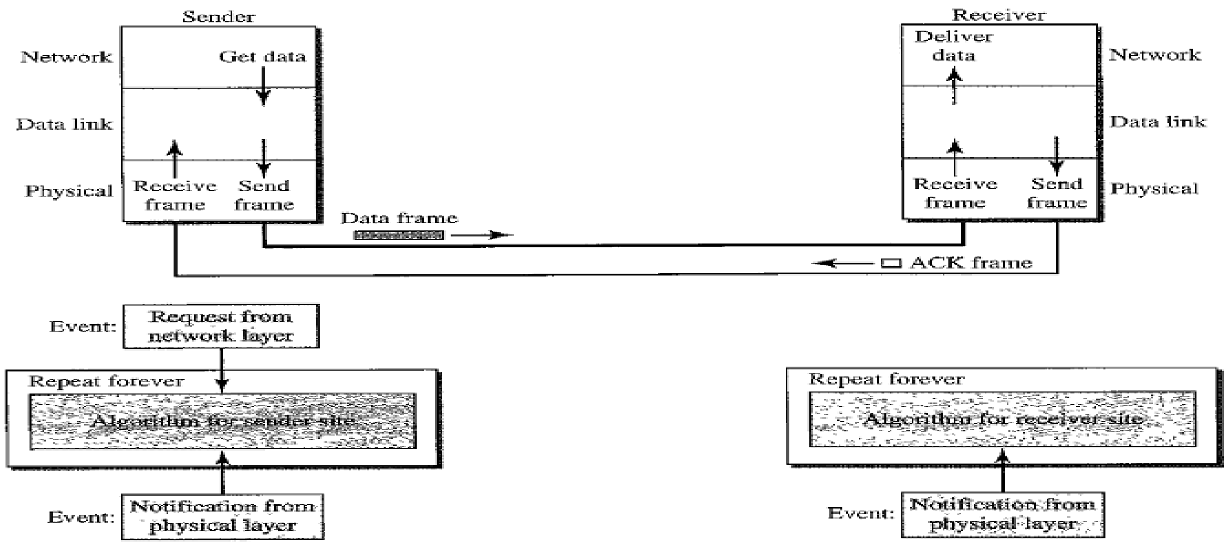
- The sender sends a sequence of frames without even thinking about the receiver.
- To send three frames, three events occur at the sender site and three events at the receiver site.

Stop-and-Wait Protocol

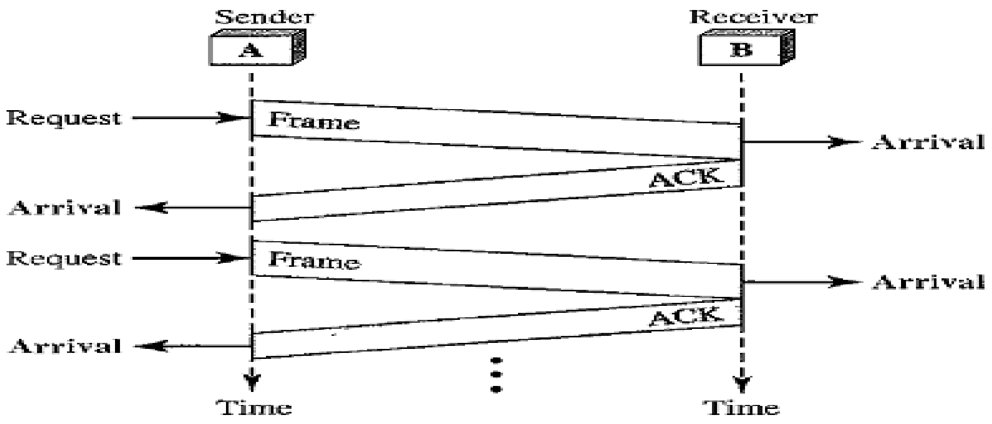
- Stop-and-Wait Protocol implements **Flow Control** Mechanism.
- If there are multiple senders sending data to one receiver, the receiver receiving data rate is slower than the sender's data rate. Then receiver is overwhelmed by the sender's data.
- To prevent the receiver from becoming overwhelmed with frames, the receiver needs to tell the sender to slow down. (i.e.) There must be feedback from the receiver to the sender.
- In Stop-and-Wait Protocol the sender sends one frame, stops until it receives confirmation (ACK's) from the receiver and then sends the next frame.

Design

At any time, there is either one data frame on the forward channel or one ACK frame on the reverse channel. Therefore need a half-duplex link.



Flow Diagram



The sender sends one frame and waits for feedback from the receiver. When the ACK arrives, the sender sends the next frame.

Sender site:
Here two events can occur:

1. **Request Event**: At the sender site the request event comes from the Network layer to Data link layer, whether Sender wants to receive the packet from the network layer.
2. **Arrival Notification**: At the sender site the arrival notification event comes from the physical layer to Data link layer.

After a frame is sent, the data link layer must ignore another network layer request until that frame is acknowledged.

- If we are designing error free frames then two arrival events cannot happen one after another.
- The requests from the network layer may happen one after another without an arrival event in between. We need somehow to prevent the immediate sending of the data frame.
- **canSend( )** method is used for this purpose, it may be either true or false.
- When a frame is sent, the variable is set to false to indicate that a new network request cannot be sent until *canSend* is true.

- When an ACK is received, **canSend** is set to true to allow the sending of the next frame.

Receiver-site
There is only one event can happen at the receiver site is the **Arrival Notification** from the Physical layer to indicate that a frame is arrived from the sender site.

After the data frame arrives, the receiver sends an ACK frame to acknowledge the receipt and allow the sender to send the next frame.

Disadvantages of Stop and Wait Protocol
1. If the receiver does not respond when there is an error then sender doesn’t know which frame has to be resend.
2. In Stop and Wait protocol the sender doesnot send next frame until it receives ACK from receiver. If the ACK has lost then sender does not know when to send the frame.

NOISY CHANNELS
Noisy means the data may be corrupted or lost.
In Noisy Channels we need to implement both Flow control and Error Control Mechanisms in the Protocols.

Stop-and-Wait Automatic Repeat Request (ARQ)
This Protocol uses Acknowledgements and Sequence Numbers to implement Flow and Error Control mechanisms. The corrupted and lost frames need to be resent in this protocol.

There are 2 Questions arises:

1. **If the receiver does not respond when there is an error, how the sender knows which frame to resend?**

Solution: The sender keeps a copy of the sent frame. At the same time, it starts a timer.

If the timer expires and there is no ACK for the sent frame then **“3 actions”** to be performed

i. The frame is resent
ii. The copy is held
iii. The timer is restarted.

Note: Since the protocol uses the stop-and-wait mechanism, there is only one specific frame that needs an ACK even though several copies of the same frame can be in the network.

2. **What if an ACK frame is also lost?**

Solution:
Since an ACK frame can also be corrupted and lost, it needs a **Sequence Number** and Redundancy Bits.

The ACK frame for this protocol has a sequence number field.

In this protocol, the sender simply discards a corrupted ACK frame or ignores an out-of-order one.

Sequence Numbers
- Sequence Number is the number that is given to the data frame.
- A field is added to the data frame to hold the sequence number of that frame.

- The sequence numbers are based on modulo-2 arithmetic.

Range: Consider the field is m bits long, the range of sequence numbers are from 0 to 2^m-1, and then the same number are repeated.

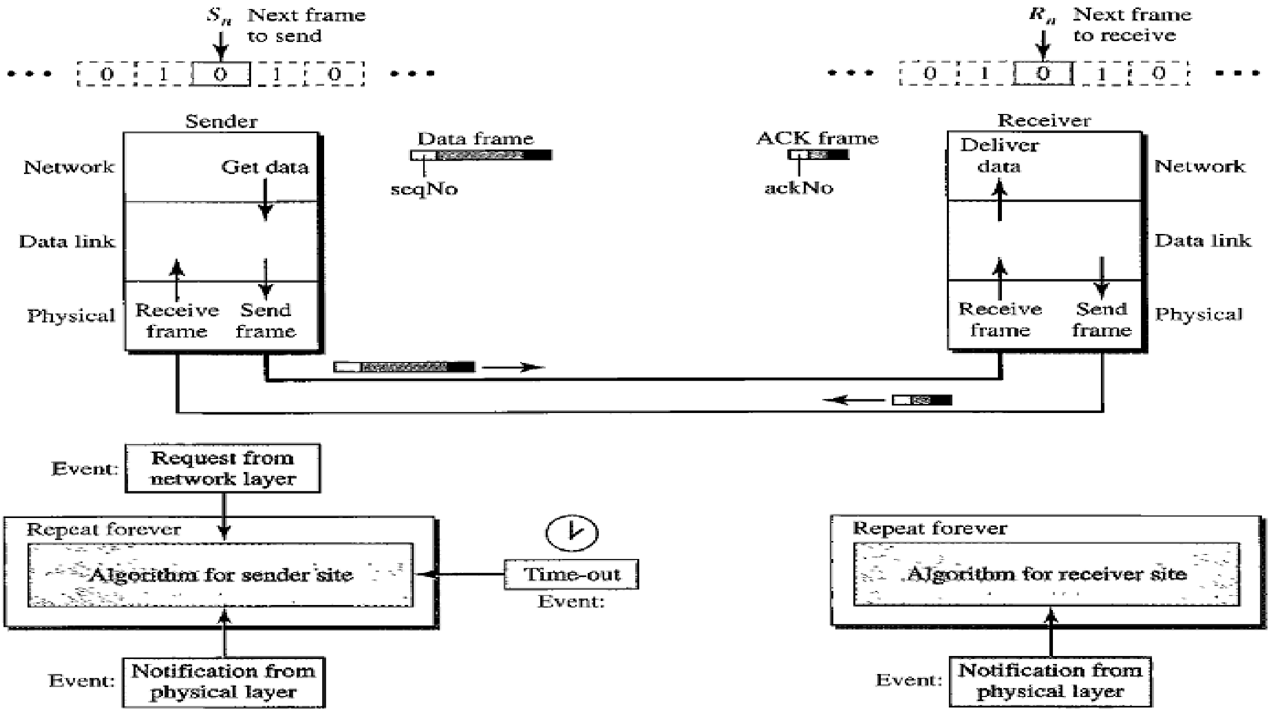
Acknowledgment Numbers

The acknowledgment numbers always announce the sequence number of the next frame expected by the receiver.

Example:

1. If frame 0 has arrived safe and sound, the receiver sends an ACK frame with acknowledgment 1 (meaning frame 1 is expected next).
2. If frame 1 has arrived safe and sound, the receiver sends an ACK frame with acknowledgment 0 (meaning frame 0 is expected).

Design



- The sending device keeps a copy of the last frame transmitted until it receives an acknowledgment for that frame.
- The design contains:
  - **seqNo**: Sequecne Number
  - **ackNo**: Acknowledgement Number
  - **S_n**: Sender Control variable- that holds the sequence number for the next frame to be sent (0 or 1).
  - **R_n**: Receiver Control Variable- that holds the number of the next frame expected.
- When a frame is sent, the value of **S_n** is incremented (modulo-2), which means if it is 0, it becomes 1 and vice versa.
- When a frame is received, the value of **R_n** is incremented (modulo-2), which means if it is 0, it becomes 1 and vice versa.

- **S_n** variable points to the slot that matches the sequence number of the frame that has been sent, but not acknowledged.
- Sequence numbers are modulo-2, the possible numbers are 0,1,0,1,0,1,……and so on.
- **R_n** points to the slot that matches the sequence number of the expected frame.
- Three events can happen at the sender site; one event can happen at the receiver site.

There are 3 Events can happen at Sender site.

1. RequestToSend (which comes from network layer to data link layer)
2. ArrivalNotification (which comes from physical layer to data link layer)
3. TimeOut

Event 1: Request to send

- Event RequestToSend is used to send the frame to receiver.
- Before the frame is sent, it is stored in the buffer. We need at least one buffer to hold this frame until sender make sure that it is received safe and sound.
- The copy is used for resending a corrupt or lost frame.
- Sender has to prevent the network layer from making a request before the previous frame is received safe and sound.

Event 2: Arrival Notification

- If the frame is not corrupted then three actions will be done:
  1. The ackNo of the ACK frame matches the sequence number of the next frame to send.
  2. Timer is stopped
  3. Purge the copy of the data frame that was saved.
- If the frame is corrupted then it ignore arrival notification event and wait for the next event to happen.

Event 3: Time Out

- After each frame is sent, a timer is started.
- When the timer expires the frame is resent and the timer is restarted.

<u>Receiver-site:</u>

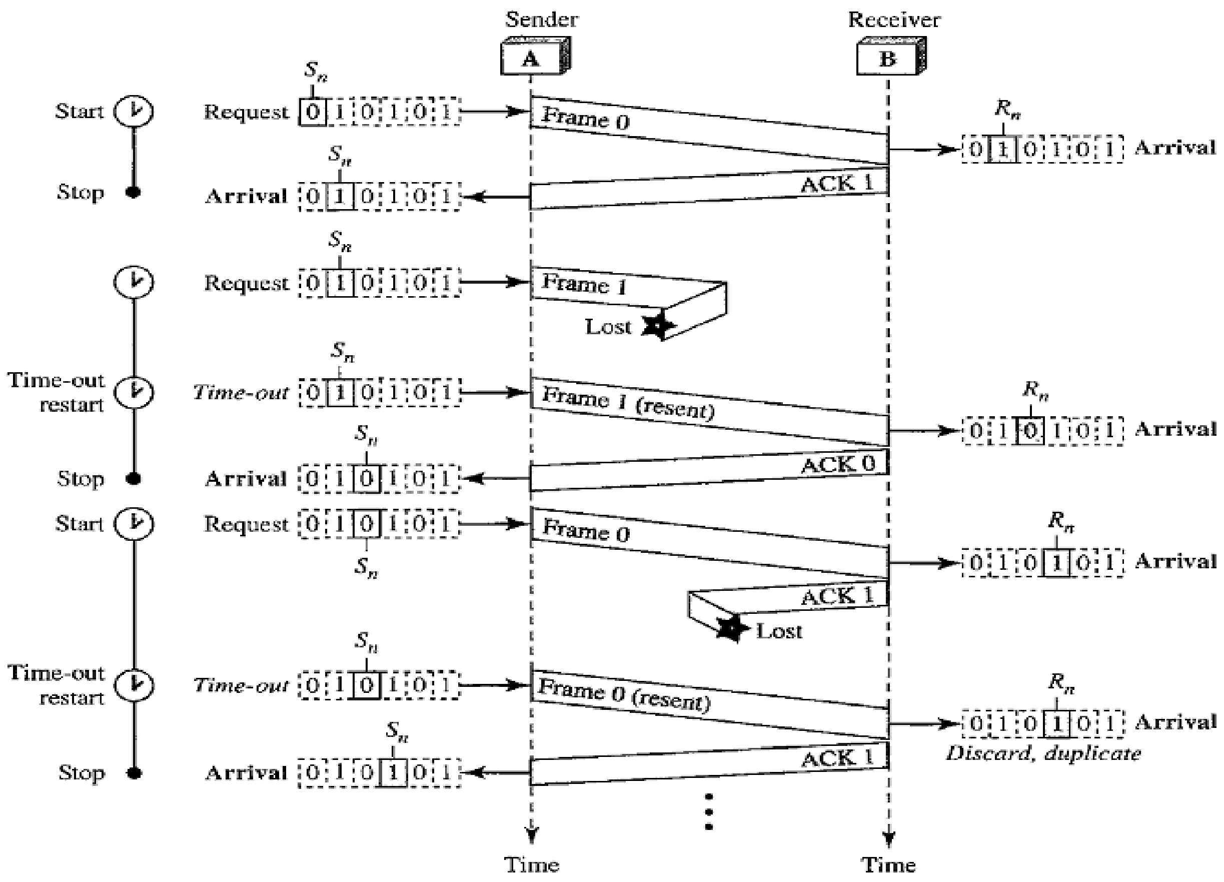
Only **One Event** happens at the receiver site.

- First, all arrived data frames that are corrupted are ignored.
- If the seqNo of the frame is the one that is expected (**R_n**), the frame is accepted, the data are delivered to the network layer, and the value of **R_n** is incremented.

Note:

- Even if the sequence number of the data frame does not match the next frame expected, an ACK is sent to the sender.
- This ACK reconfirms the previous ACK instead of confirming the frame received.
- This is done because the receiver assumes that the previous ACK might have been lost; the receiver is sending a duplicate frame.
- The resent ACK may solve the problem before the time-out does it.

Flow diagram



- Frame 0 is sent and acknowledged.
- Frame 1 is lost and resent after the time-out.
- The resent frame 1 is acknowledged and the timer stops.
- Frame 0 is sent and acknowledged, but the acknowledgment is lost.
- The sender has no idea if the frame or the acknowledgment is lost, so after the time-out, it resends frame 0, which is acknowledged.

Pipelining

- In networking a task is often begun before the previous task has ended. This is known as pipelining.
- There is no pipelining in Stop-and-Wait ARQ because sender need to wait for a frame to reach the destination and be acknowledged before the next frame can be sent.
- Go-Back-N ARQ and Selective Repeat ARQ use Pipelining, because several frames can be sent before sender receives the ACK about previous frame.
- Pipelining improves the efficiency of the transmission if the number of bits in transition is large with respect to the bandwidth-delay product.

Go-Back-N Automatic Repeat Request (ARQ)

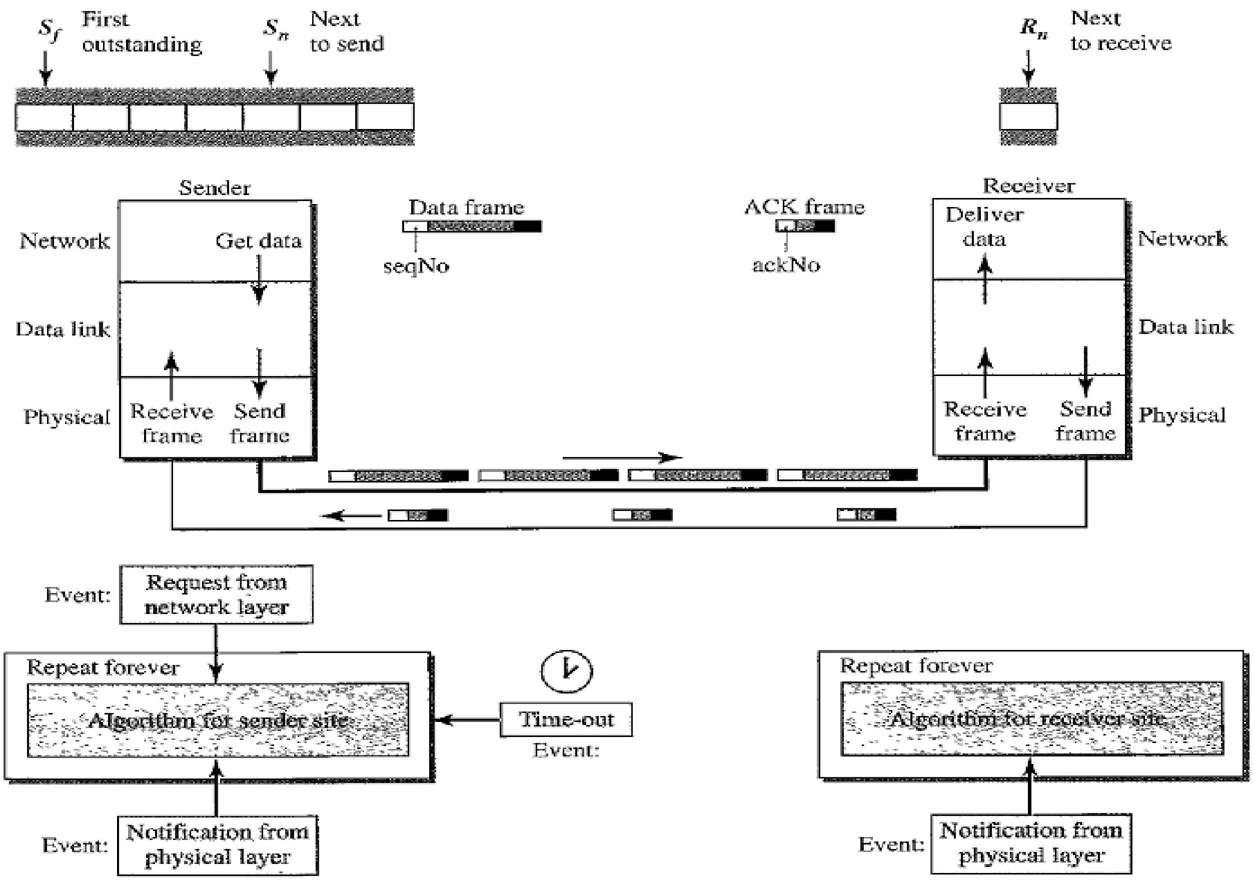
In this protocol

- 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

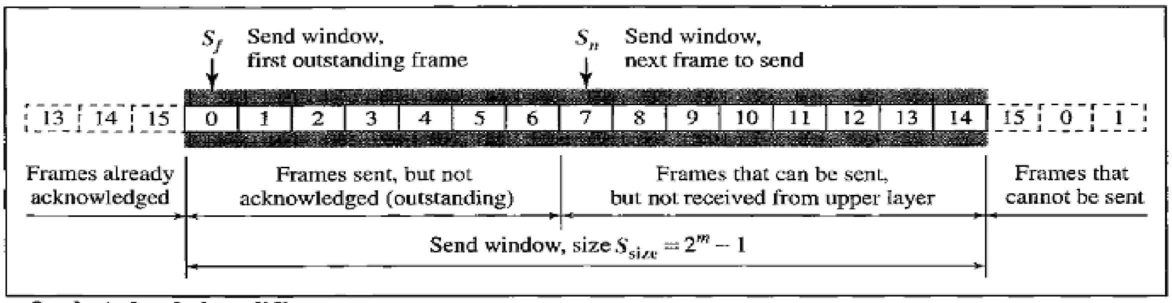
The range that is the concern of receiver is called receive sliding window or receive window.

Send Window

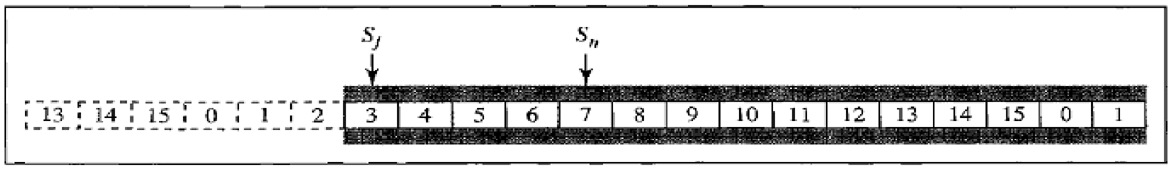
- The send window is an imaginary box covering the sequence numbers of the data frames which can be in transit.
- In each window position, some of these sequence numbers define the frames that have been sent; others define those that can be sent.
- The maximum size of the send window is **2^m – 1.**

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

Consider the below figure:


S_f Send window, first outstanding frame
S_n Send window, next frame to send
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1
Frames already acknowledged
Frames sent, but not acknowledged (outstanding)
Frames that can be sent, but not received from upper layer
Frames that cannot be sent
Send window, size $S_{size} = 2^m - 1$

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.

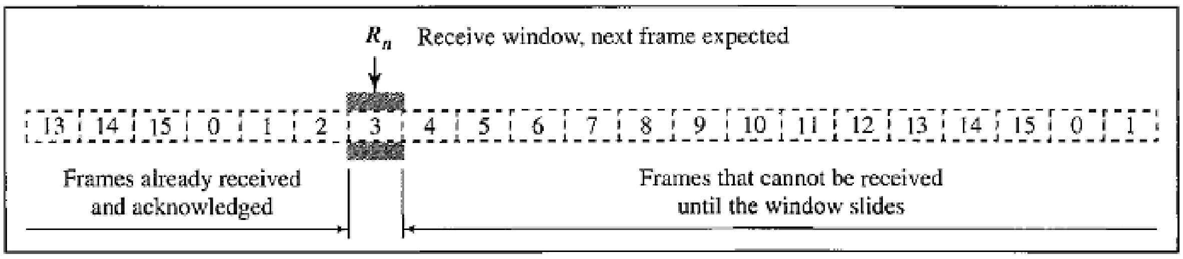
Receive Window

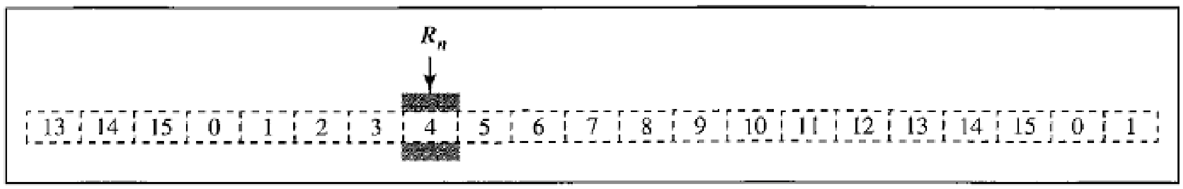
- The receive window makes sure that the correct data frames are received and that the correct acknowledgments are sent.
- The size of the receive window is always 1.
- The receiver is always looking for the arrival of a specific frame.
- Any frame arriving out of order is discarded and needs to be resent.

There is only one variable required, that is $\mathbf{R_n}$

$\mathbf{R_n}$: 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 $\mathbf{R_n}$ is accepted and acknowledged. (i.e. $\mathbf{S_n=R_n}$)
- The receive window slides only one slot at a time, when the correct frame is received the window slides.


R_n
Receive window, next frame expected
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1
Frames already received and acknowledged
Frames that cannot be received until 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:

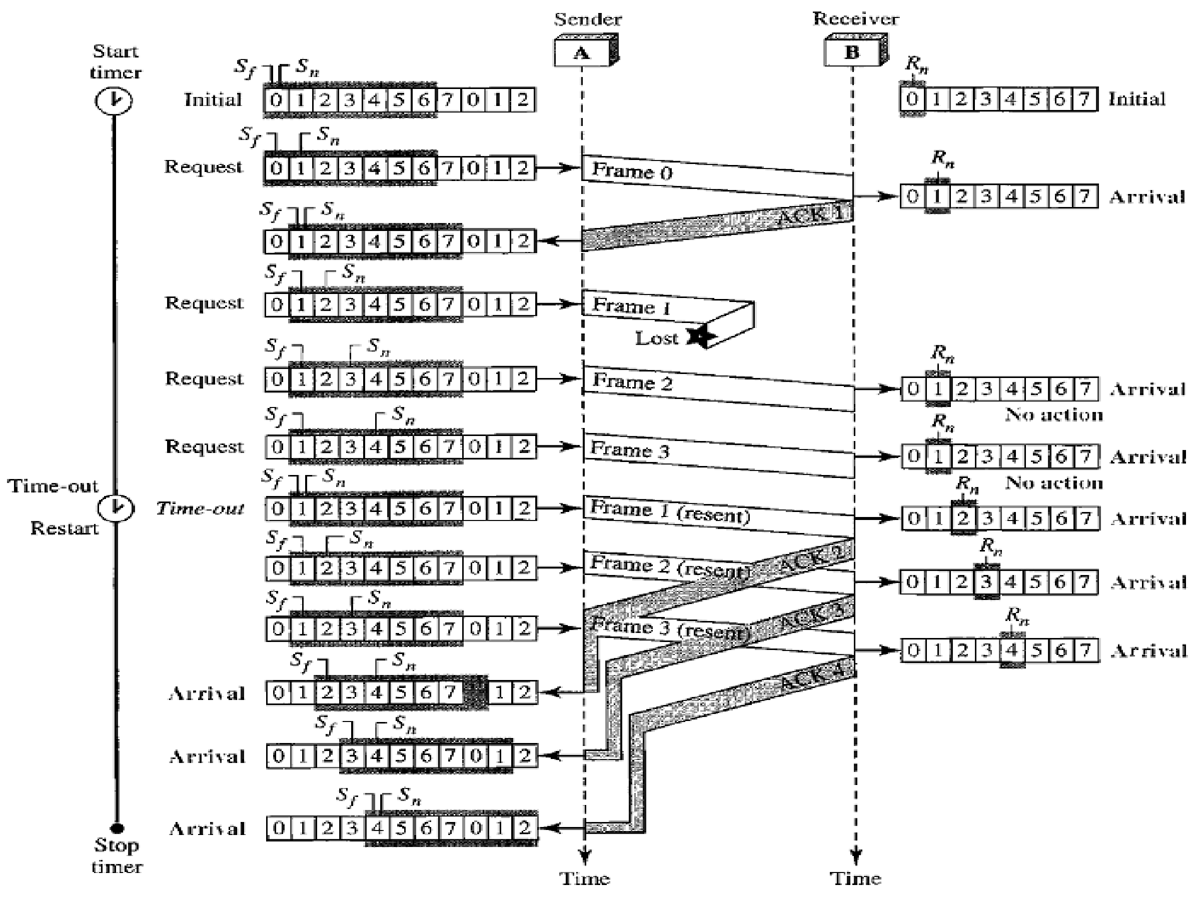
- Suppose the sender has already sent frame 6, but the timer for frame 3 expires.
- This means that frame 3 has not been acknowledged.
- The sender goes back and sends frames 3, 4,5, and 6 again.
- That is why the protocol is called ***Go-Back-N*** **ARQ**.

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.

SELECTIVE REPEAT Automatic Repeat Request

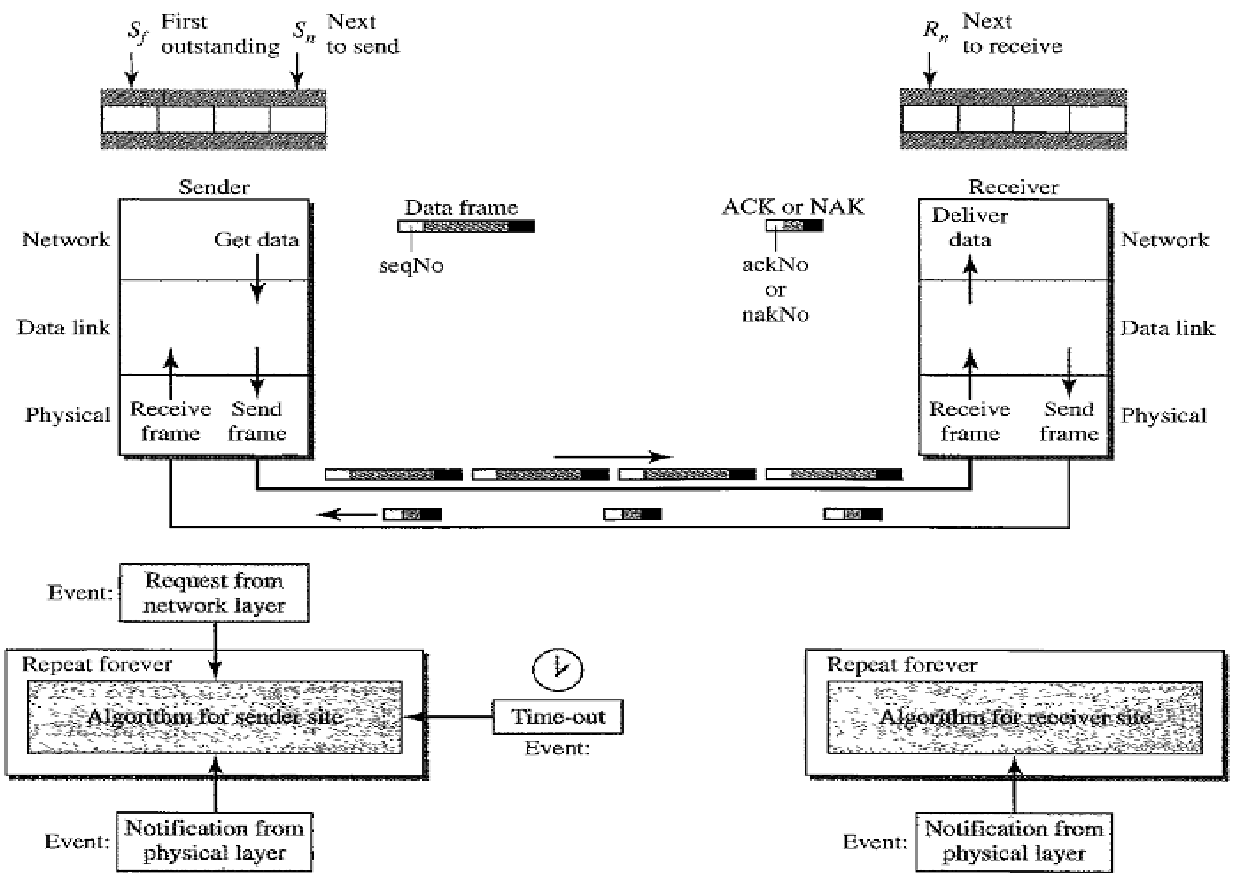
It is more efficient for Noisy links but processing at the receiver is more complex.

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).



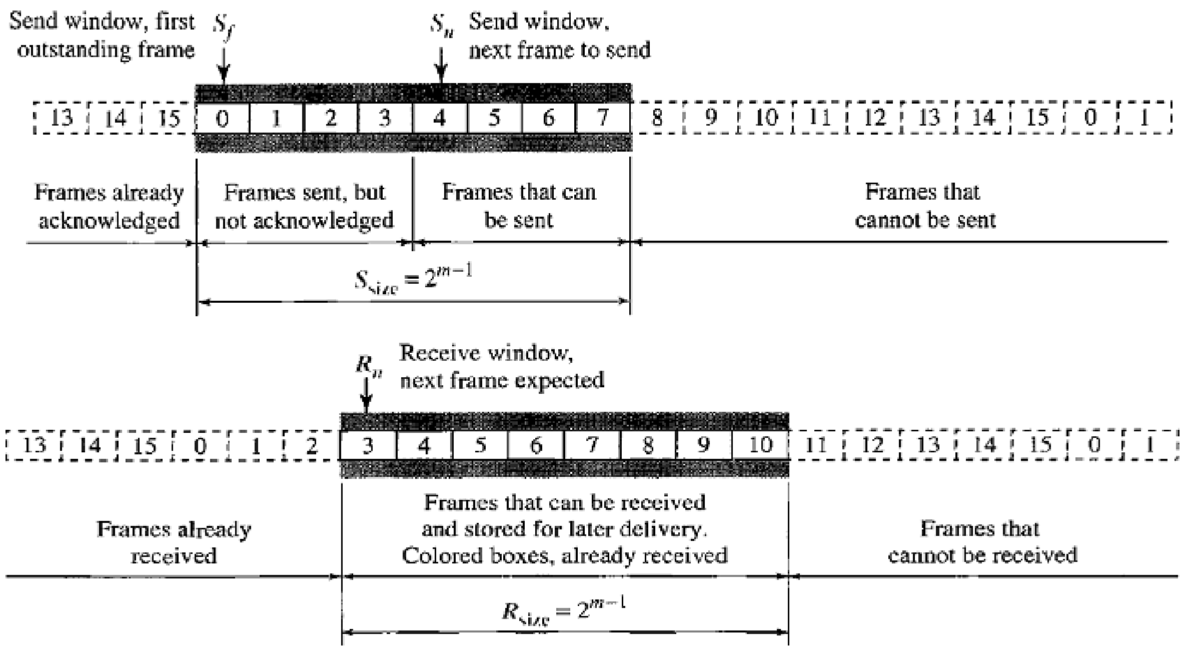
Windows

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.


Send window, first outstanding frame S_f
S_n Send window, next frame to send
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1
Frames already acknowledged
Frames sent, but not acknowledged
Frames that can be sent
Frames that cannot be sent
$S_{size} = 2^{m-1}$
R_n Receive window, next frame expected
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1
Frames already received
Frames that can be received and stored for later delivery. Colored boxes, already received
Frames that cannot be received
$R_{size} = 2^{m-1}$

- 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:

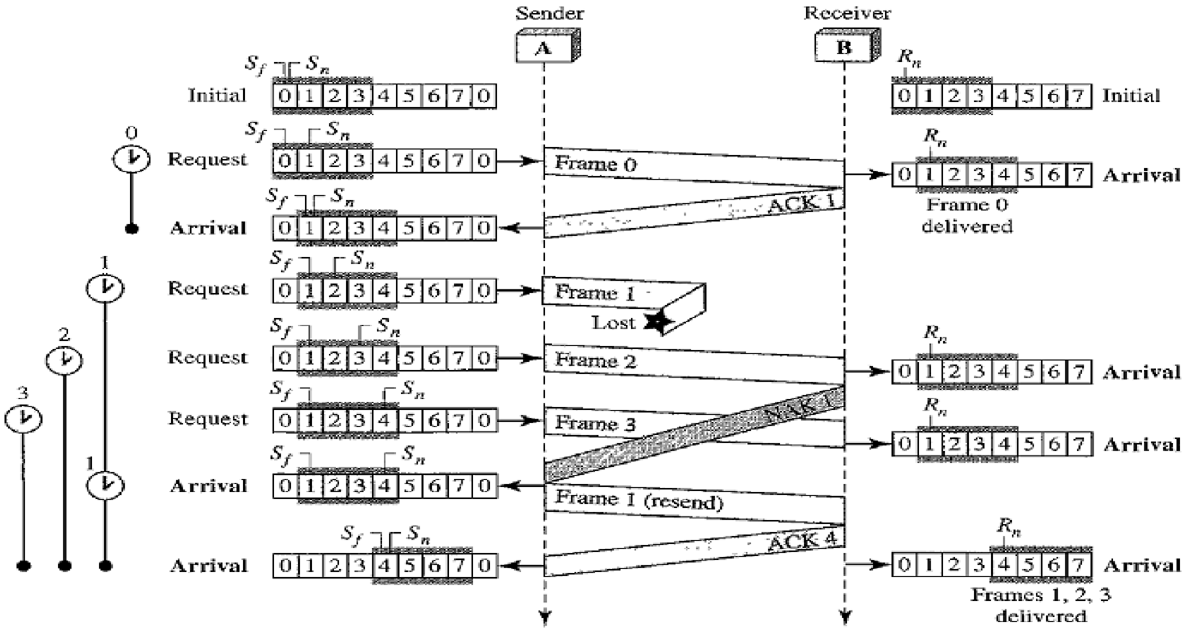
- For every request event a timer is started before sending any frame. The arrival event is more complicated. An ACK or a NAK frame may arrive at the sender site.
- If a valid NAK frame arrives, we just resend the corresponding frame.
- If a valid ACK arrives 3 actions will be done:
  1. Purge the buffers
  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 :

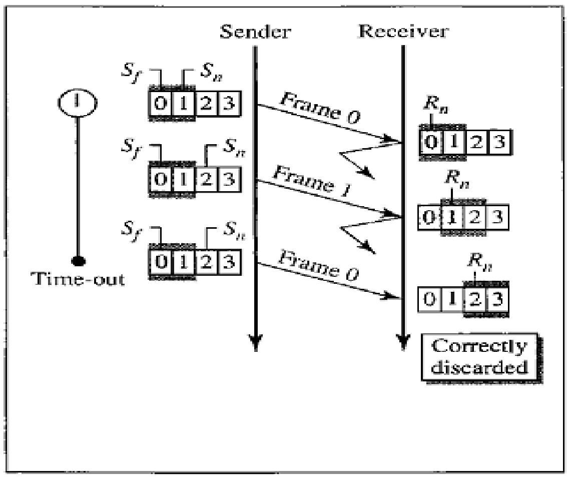
Why the size of the sender and receiver windows is 2^{m-1}?

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

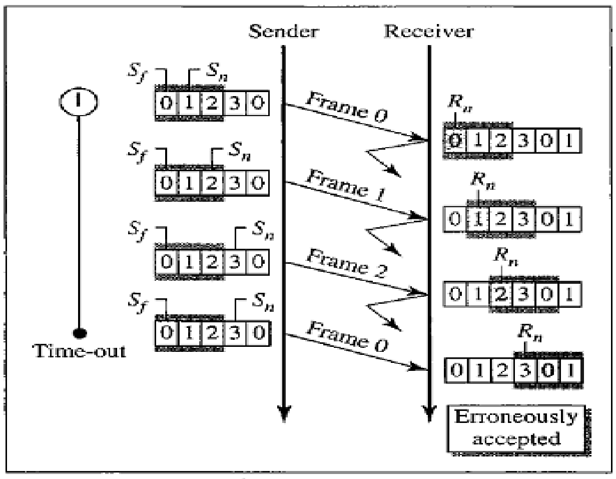
Now we compare window size=2 and window size =3.

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.

PIGGYBACKING

- The protocols Stop-and-wait ARQ, Go-Back-N ARQ, Selective Repeat ARQ are all unidirectional. That means data frames flow in only one direction although control information such as ACK and NAK frames can travel in the other direction.
- In real life, data frames are normally flowing in both directions: from node A to node B and from node B to node A. This means that the control information also needs to flow in both directions. This technique called **piggybacking.**
- **Piggybacking** is used to improve the efficiency of the bidirectional protocols. When a frame is carrying data from A to B, it can also carry control information from B to A. Similarly when a frame is carrying data from B to A, it can also carry control information frames from A to B.

HIGH-LEVEL DATA LINK CONTROL (HDLC)

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

HDLC provides two common transfer modes that can be used in different configurations:

1. Normal Response Mode (NRM)
2. Asynchronous Balanced Mode (ABM)

Normal Response Mode (NRM)	Asynchronous Balanced Mode
1. In normal response mode (NRM), the station configuration is unbalanced. 2. We have one primary station and multiple secondary stations. 3. A primary station can send commands; a secondary station can only respond. 4. The NRM is used for both point-to-point and multiple-point links.	1. In asynchronous balanced mode (ABM), the configuration is balanced. 2. Each station can function as a primary and a secondary station. 3. Both Primary and Secondary stations can send Commands and Responces. 4. The link is point-to-point.

Frames

HDLC defines three types of frames, each type of frame serves as an envelope for the transmission of a different type of message.

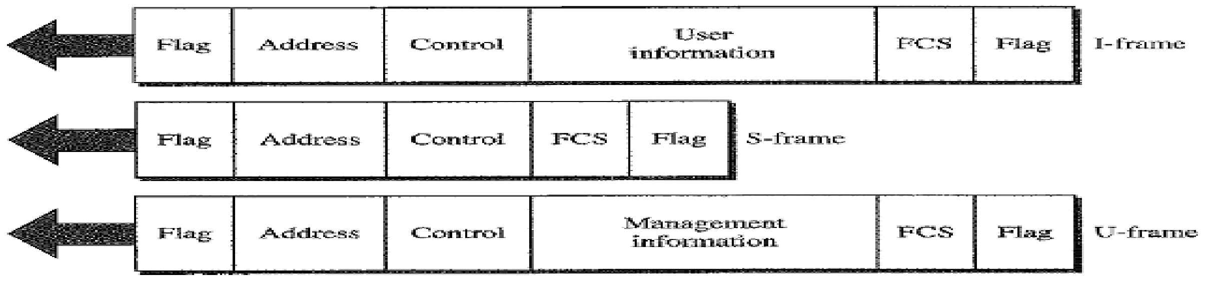
i. **Information frames (I-frames)**
 I-frames are used to transport user data and control information relating to user data. Only I-frames uses piggybacking.

ii. **Supervisory frames (S-frames)**
 S-frames are used only to transport control information (i.e. ACK,NAK)

iii. **Unnumbered frames (U-frames)**
 U-frames are reserved for system management. Information carried by U-frames is intended for managing the link itself. U-frames do not have sequence numbers.

Frame Format

The below figure describes different frames and their fields where as a beginning flag field, an address field, a control field and an ending flag field is common in all frames.



Flag field

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

Address field

- The second field of an HDLC frame contains the address of the secondary station.

- If a primary station created the frame, it contains a "*to* address".
- If a secondary creates the frame, it contains *a "from* address".
- An address field can be 1 byte or several bytes long, depending on the needs of the network. One byte (8-bits) can identify up to 128 stations ($2^7=2^{8-1}$).
- Larger networks require multiple-byte address fields.
- If the address field is only 1 byte, the last bit is always a "1".
- If the address is more than 1 byte, all bytes but the last one will end with 0; only the last will end with 1.
- Ending each intermediate byte with 0 indicates to the receiver that there are more address bytes to come.

Control field

- The control field is a 1-byte or 2-byte segment of the frame used for flow and error control.

Information field

- The information field contains the user's data from the network layer or management information. Its length can vary from one network to another.

FCS field

- Frame Check Sequence (FCS) is the HDLC error detection field.
- FCS contains either a 2-byte or 4-byte ITU-T CRC (i.e CRC-16 or CRC-32).

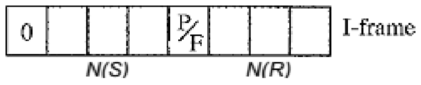
<u>Control Field</u>

The control field determines the type of frame and defines its functionality.

If the first 2 bits of control field determines the type:

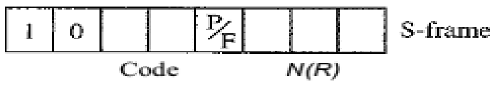
- If first bit is 0, then it is I-Frame.
- If first 2 bits are 10, then it is S-Frame.
- If first 2 bits are 11, then it is U-Frame.

Control Field for I-Frames



- The first bit is 0, that defines it is I-Frame.
- I-frames are designed to carry user data from the network layer.
- I-Frames can include flow and error control information (piggybacking). The subfields in the control field are used to define these functions.
- **N(S)** is a 3 bit sequence number of a frame occupies 2nd, 3rd, 4th bits. The range of sequence numbers are 0-7.
- **N(R)** is 3 bit acknowledgement number occupies last 3 bits (i.e. 6th, 7th, 8th bits) used for Piggybacking.
- **P/F (Poll/Final) bit** is the 5th bit, it is a single bit with dual purpose, and it has to set to value'1'.
- **poll** means when the frame is sent by a primary station to a secondary, (i.e.) when the address field contains the address of the receiver.
- **final** means when the frame is sent by a secondary to a primary, (i.e.)when the address field contains the address of the sender.

Control Field for S-Frames



- If the first 2 bits of the control field is 10, this means the frame is an S-frame.
- Supervisory frames are used for flow and error control whenever piggybacking is either impossible or inappropriate.
- S-frames do not have information fields.
- The last 3 bits, called *N(R),* corresponds to the acknowledgment number (ACK) or negative acknowledgment number (NAK) depending on the type of S-frame.

The 2 bits called code is used to define the type of S-frame,

Receive Ready (RR) S-frame (00)

- RR S-frame acknowledges the receipt of a safe and sound frame or group of frames.
- The value ***N(R)*** field defines the Acknowledgment Number (ACK).

Receive Not Ready (RNR) S-Frame (10)

- It acknowledges the receipt of a frame or group of frames, and it announces that the receiver is busy and cannot receive more frames.
- It acts as a kind of congestion control mechanism by asking the sender to slow down.
- The value of ***N(R)*** is the Acknowledgment Number (ACK).

Reject (REJ) S-Frame (01)

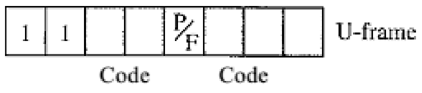
- It is a NAK frame that can be used in *Go-Back-N* ARQ to improve the efficiency of the process by informing the sender, before the sender time expires, that the last frame is lost or damaged.
- The value of ***N(R)*** is the Negative Acknowledgment Number (NAK).

Selective Reject (SREJ) S-Frame (11)

- This is a NAK frame used in Selective Repeat ARQ.
- The value of ***N(R)*** is the Negative Acknowledgment Number (NAK).

S-Frame Type	**Code**	**N(R)**	**Usage**
RR	00	ACK	Acknowledge the receipt of safe & sound frame.
RNR	10	ACK	Informs sender that Receiver is busy.
REJ	01	NAK	Used in Go-Back-N ARQ.
SREJ	11	NAK	Used in Selective Repeat ARQ.

Control Field for U-Frames



- Unnumbered frames are used to exchange session management and control information between connected devices.
- U-frames contains Information field, but one used for system management information, not user data.
- U-frame codes are divided into two sections: (2+3=5 bits)
  1. A 2-bit prefix before the P/F bit
  2. 3-bit suffix after the P/F bit

- Together, these two segments (5 bits) can be used to create up to 32 different types of U-frames.

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

POINT-TO-POINT PROTOCOL (PPP)

Point-to-Point Protocol one of the most common protocols for point-to-point access.

Millions of Internet users who need to connect their home computers to the server of an Internet service provider use PPP.

The majority of these users have a traditional modem; they are connected to the Internet through a telephone line, which provides the services of the physical layer. But to control and manage the transfer of data, there is a need for a point-to-point protocol at the data link layer.

PPP-Services

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

Limitations of PPP

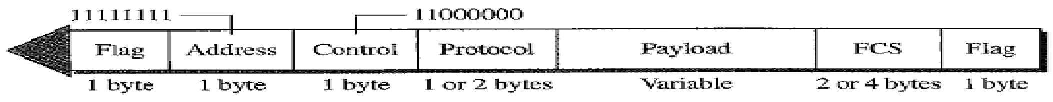
- **No Flow control**
  PPP does not provide flow control. A sender can send several frames one after another with no concern about overwhelming the receiver.
- **Lack of error control**
  PPP has a very simple mechanism for error control. A CRC field is used to detect errors. If the frame is corrupted, it is silently discarded; the upper-layer protocol needs to take

care of the problem. Lack of error control and sequence numbering may cause a packet to be received out of order.

- **No Effective Addressing**
  PPP does not provide a sophisticated addressing mechanism to handle frames in a multipoint configuration.

Framing

PPP is a byte-oriented protocol. Frame Format consisting of: Flag, Address, Control, Protocol, Payload, FCS.



Flag
- A PPP frame starts and ends with a 1-byte flag with the bit pattern 01111110.
- PPP is a byte-oriented protocol. The flag is treated as a byte.

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

Control
- This field is set to the constant value 11000000.
- PPP does not provide any flow control.
- Error control is also limited to error detection.
- This field byte is omitted after negotiation of two parties.

Protocol
- The protocol field defines what is being carried in the data field: either user data or other information such as ACK,NAK etc.
- The length of the field is either 1 or 2-bytes.

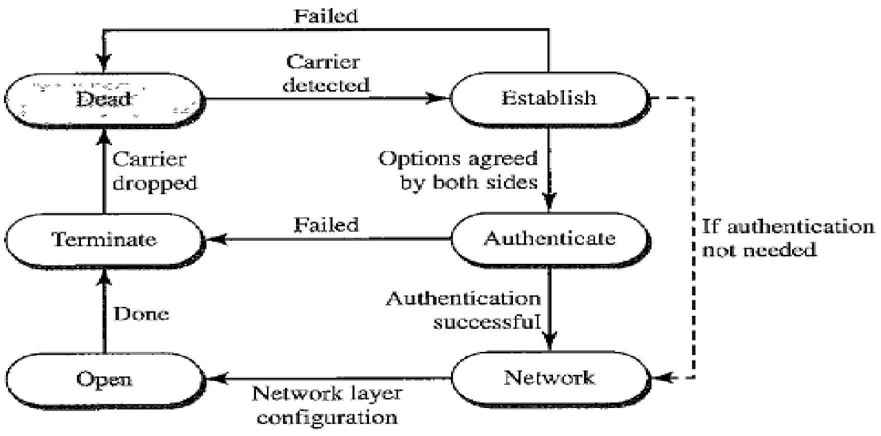
Payload field
- This field carries either the user data or other information.
- The data field is a sequence of bytes with the default of a maximum of 1500 bytes;
- The data field is byte stuffed if the flag byte pattern appears in this field. Because there is no field defining the size of the data field, padding is needed if the size is less than the maximum default value or the maximum negotiated value.

FCS
- The frame check sequence (FCS) is simply a 2-byte or 4-byte standard CRC.

Byte Stuffing
- PPP is a byte-oriented protocol whereas HDLC is a bit oriented protocol.
- The flag in the PPP is a byte; it needs to be escaped whenever it appears in the data section of the frame.
- Every time the flag like pattern appears in the data, the ***Escape Byte (ESC)*** is 01111101, stuffed to tell the receiver that the next byte is not a flag.

Transition Phases

A PPP connection goes through 6 transition phases:


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

Dead

- In the dead phase the link is not being used.
- There is no active carrier at the physical layer and the line is quiet.

Establish

- When one of the nodes starts the communication, the connection goes into this phase.
- In this phase, options are negotiated between the two parties (sender and receiver).
- If the negotiation is successful, the system goes to the authentication phase or directly to the networking phase.

Authenticate

- The authentication phase is optional.
- In the authentication phase the two nodes send several authentication packets.
- If the result is successful then the connection goes to the **Networking** phase;
- If result is not successful then the connection goes to the **Termination** phase.

Network

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

Open

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

Terminate

- In the termination phase the connection is terminated.
- Several packets are exchanged between the two ends for closing the link.

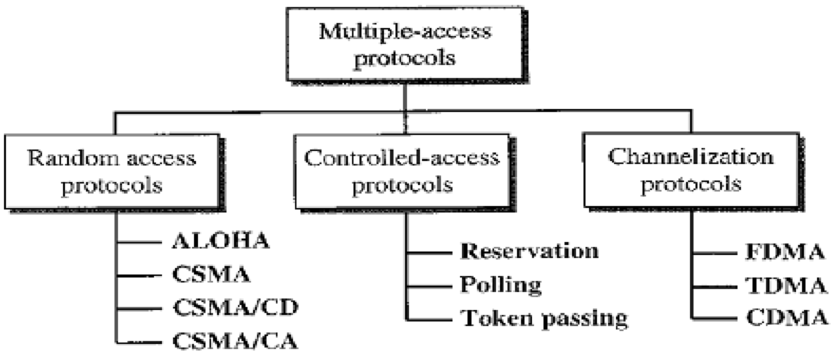
MULTIPLE ACCESS RESOLUTION PROTOCOLS

The Data Link Layer can be divided into 2 sublayers:

1. Data Link Control
2. Multiple Access Resolution

- The Multiple access resolution layers is responsible for resolving access to the shared media.
- Multiple Access Resolution layer is needed only when the channel is not dedicated. If the channel is dedicated then there is no need for multiple access sublayer.
- When nodes or stations are connected and use a common link, called a **Multipoint link** or **Broadcast link**, we need a multiple-access protocol to coordinate access to the link.
- The protocols have been devised to handle access to a shared link are called **Multiple Access Resolution Protocols**.

These protocols are categorized as below:



Random Access Protocols

Random Access method is also called Contention method.

- There is no scheduled time for a station to transmit. Transmission is random among the stations. That is why these methods are called **Random Access methods**.
- There are no rules to specify which station should send next. Stations compete with one another to access the medium. That is why these methods are also called **Contention Methods**.

At each instance, a station that has data to send, uses a procedure defined by the protocol to make a decision on whether to send the data or not to send the data. This decision depends on the state of the medium (idle or busy).

In this method:

- No station is superior to another station.
- No station is assigned the control over another station.
- No station permits or does not permit another station to send.

In this method, each station has the right to the medium without being controlled by any other station. If more than one station tries to send, there is an **Access Conflict-Collision** and the frames will be either destroyed or modified during the collision.

In order to avoid these collisions 4 protocols are implemented, they are:

- ALOHA
- CSMA
- CSMA/CD
- CSMA/CA

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

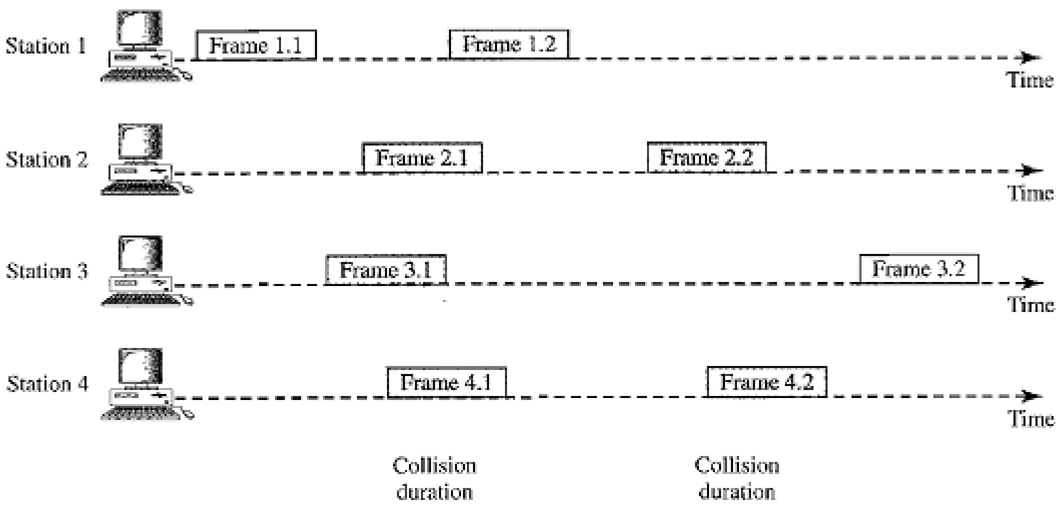
i. Pure ALOHA

ii. Slotted ALOHA

Pure ALOHA

The original ALOHA or Pure ALOHA is a simple protocol.

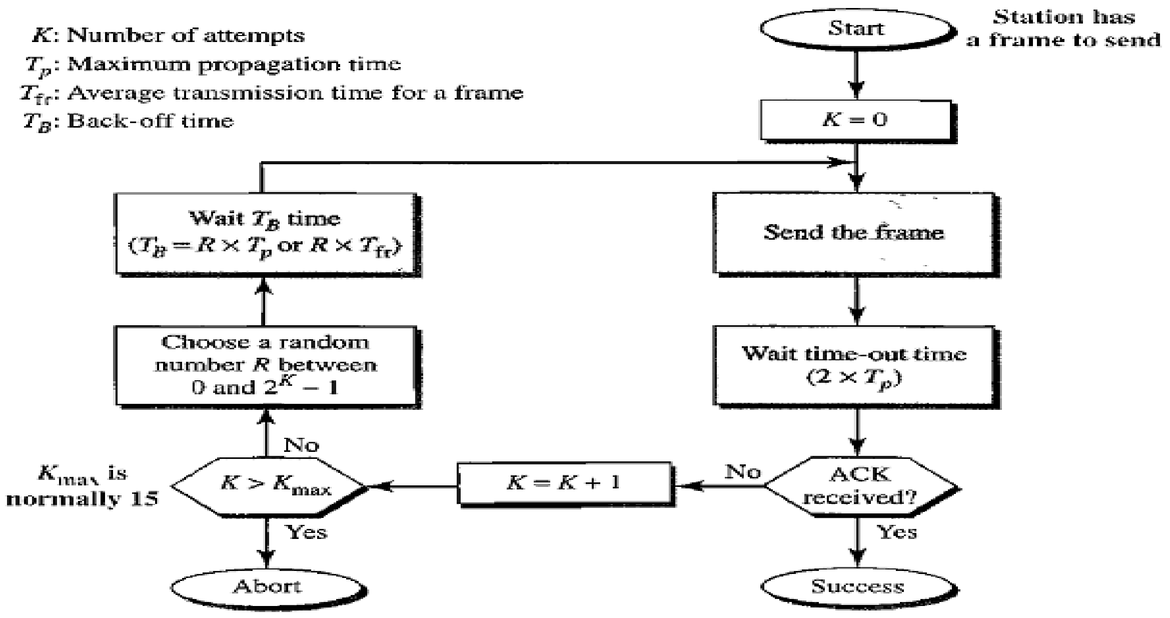
The idea is that each station sends a frame whenever it has a frame to send, there is only one channel to share, and there is the possibility of collision between frames from different stations.



- 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:



K: Number of attempts
Tp: Maximum propagation time
Tfr: Average transmission time for a frame
TB: Back-off time
Start
Station has a frame to send
K = 0
Send the frame
Wait time-out time (2 × Tp)
ACK received?
No
Yes
Success
K = K + 1
K > Kmax
Kmax is normally 15
Yes
Abort
No
Choose a random number R between 0 and 2^K − 1
Wait TB time (TB = R × Tp or R × Tfr)

- 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 $\mathbf{2^K - 1}$ is randomly chosen and multiplied by $\mathbf{T_P}$ (maximum propagation time) or $\mathbf{T_{fr}}$ (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.

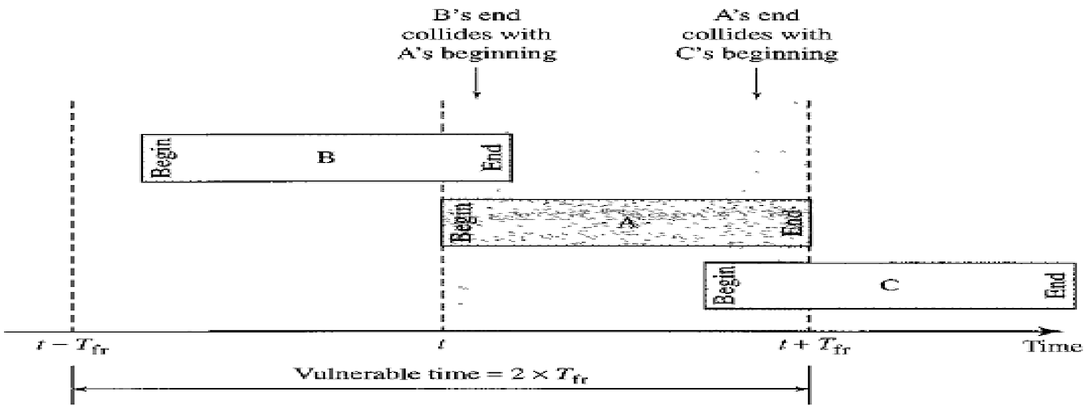
Let us assume that the stations send fixed-length frames with each frame taking $\mathbf{T_{fr}}$**'s** to send.

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

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


B's end collides with A's beginning
A's end collides with C's beginning
Begin
B
End
Begin
A
End
Begin
C
End
$t - T_{fr}$
t
$t + T_{fr}$
Time
Vulnerable time = $2 \times T_{fr}$

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 x 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.

Slotted ALOHA

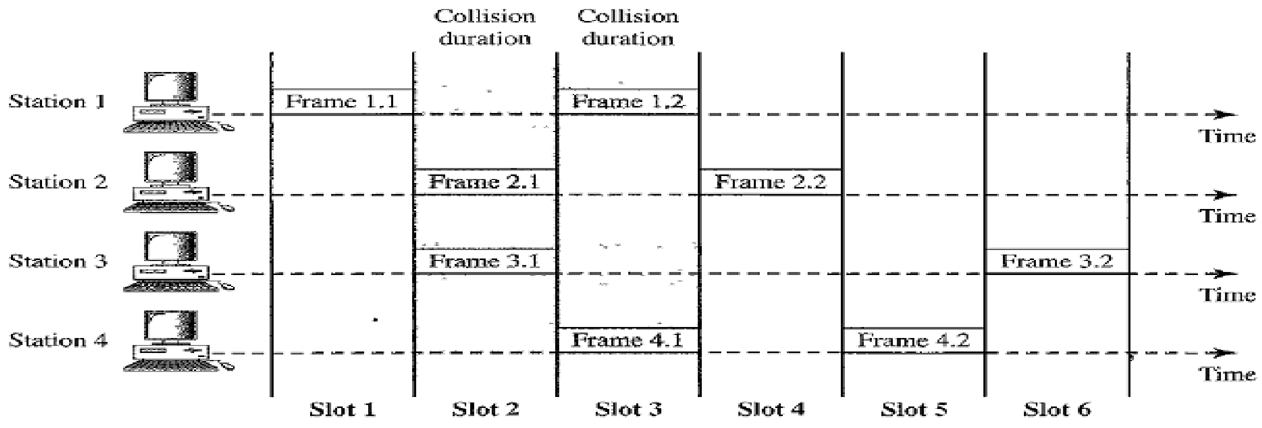
Pure ALOHA has a vulnerable time of **2 x T_{fr}**. This is so because there is no rule that defines when the station can send.

- 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.



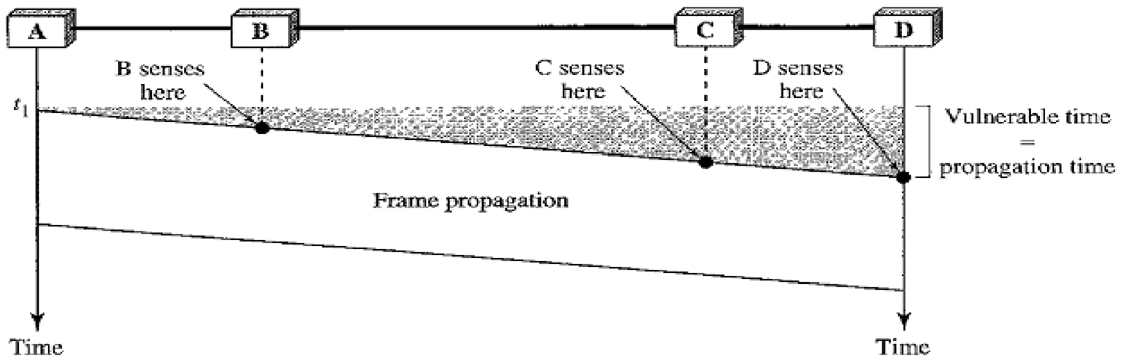
Throughput

- The throughput for slotted ALOHA is ***S* =: G x *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.

Carrier Sense Multiple Access (CSMA)

CSMA was developed to minimize the chance of collision and improve the performance, but it cannot eliminate the collision.

- CSMA is based on the principle of "sense before transmit" or "listen before talk".
- That means, CSMA requires that each station first listen to the medium or check the state of the medium before sending. Stations are connected to a shared channel.
- Collision occurs due to propagation delay. (i.e.) when a station sends a frame, it still takes time for the first bit to reach every station and for every station to sense it.
- At time t1, station B senses the medium and finds it idle, so it sends a frame.
- At time t2 (t2> t1) station C senses the medium and finds it idle because at this time, the first bits from station B have not reached station C.
- Station C also sends a frame. The two signals collide and both frames are destroyed.



Vulnerable Time of CSMA

$$\boxed{\text{Vulnerable Time= Propagation Time}}$$

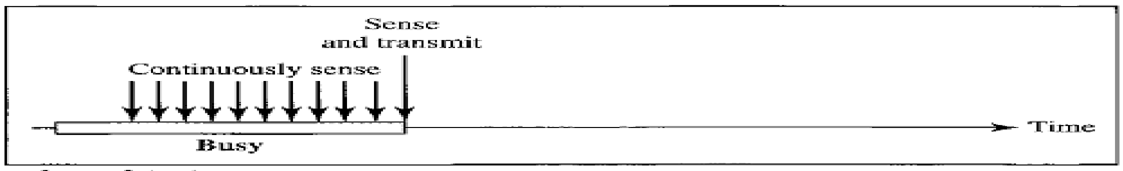
- It is the time needed for a signal to propagate from one end of the medium to the other.
- The leftmost station A sends a frame at time t1 which reaches the rightmost station D at time $t_1 + T_P$.

Persistence Methods

Persistence methods define what action will be taken by a station if the channel is busy or idle. There are 3 methods in persistence method:

1-persistent method

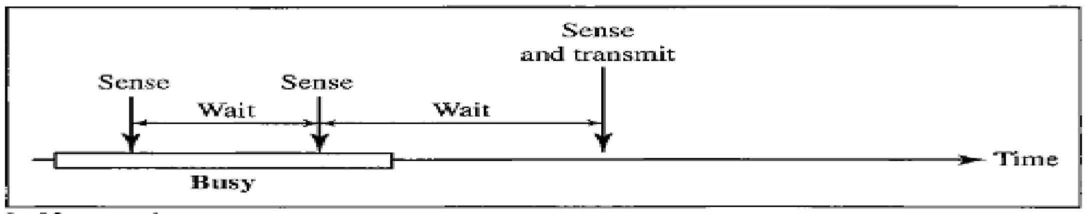
- 1-persistent method is simple and straightforward.
- In this method, after the station finds the line idle, it sends its frame immediately (with probability 1).
- This method has the highest chance of collision because two or more stations may find the line idle and send their frames immediately.



a. 1-persistent

Non-persistent Method

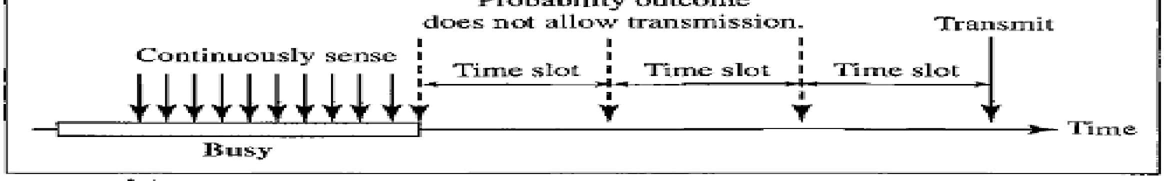
- In the non-persistent method, a station that has a frame to send senses the line. If the line is idle, it sends immediately.
- If the line is not idle, it waits a random amount of time and then senses the line again.
- The non-persistent approach reduces the chance of collision because it is unlikely that two or more stations will wait the same amount of time and retry to send simultaneously.
- However, this method reduces the efficiency of the network because the medium remains idle when there may be stations with frames to send.



b. Nonpersistent

p-Persistent Method

- The p-persistent method is used if the channel has time slots with a slot duration equal to or greater than the maximum propagation time.
- It reduces the chance of collision and improves efficiency.

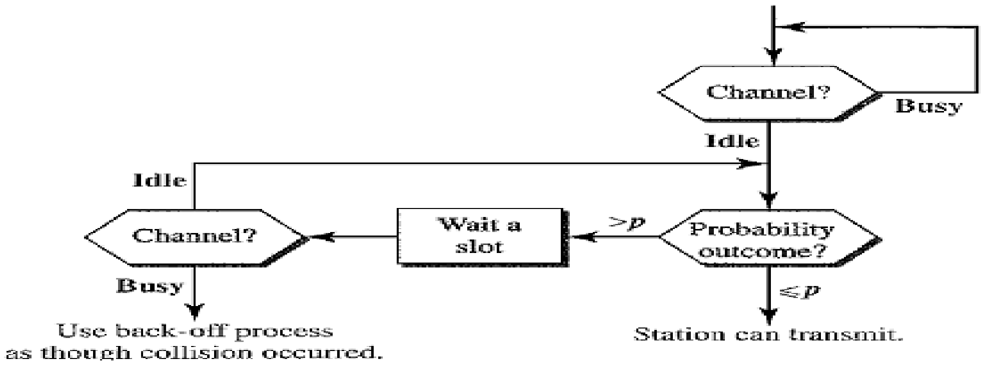


c. *p*-persistent

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

- With probability *p*, the station sends its frame.
- With probability *q* = 1 - *p*, the station waits for the beginning of the next time slot and checks the line again.
- If the line is idle, it goes to step 1.

- If the line is busy, it acts as though a collision has occurred and uses the backoff procedure.

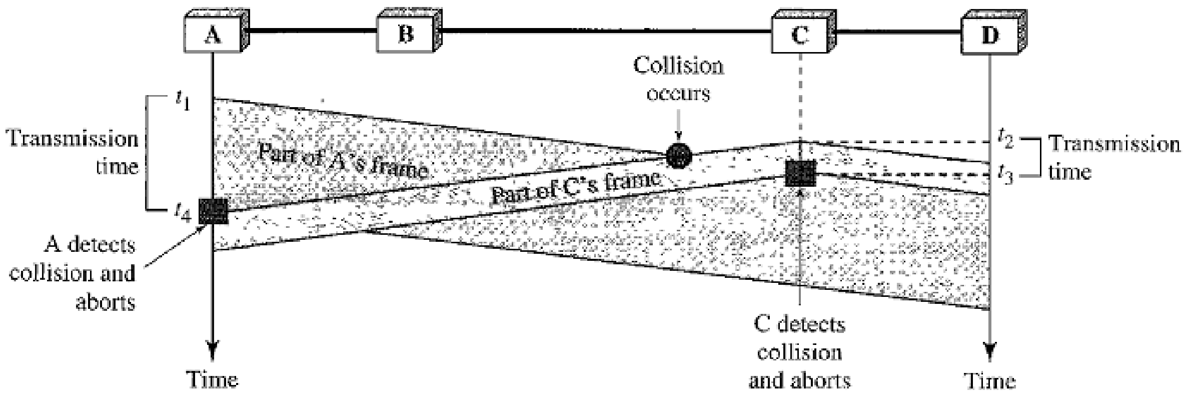


Carrier Sense Multiple Access with Collision Detection (CSMA/CD)

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.

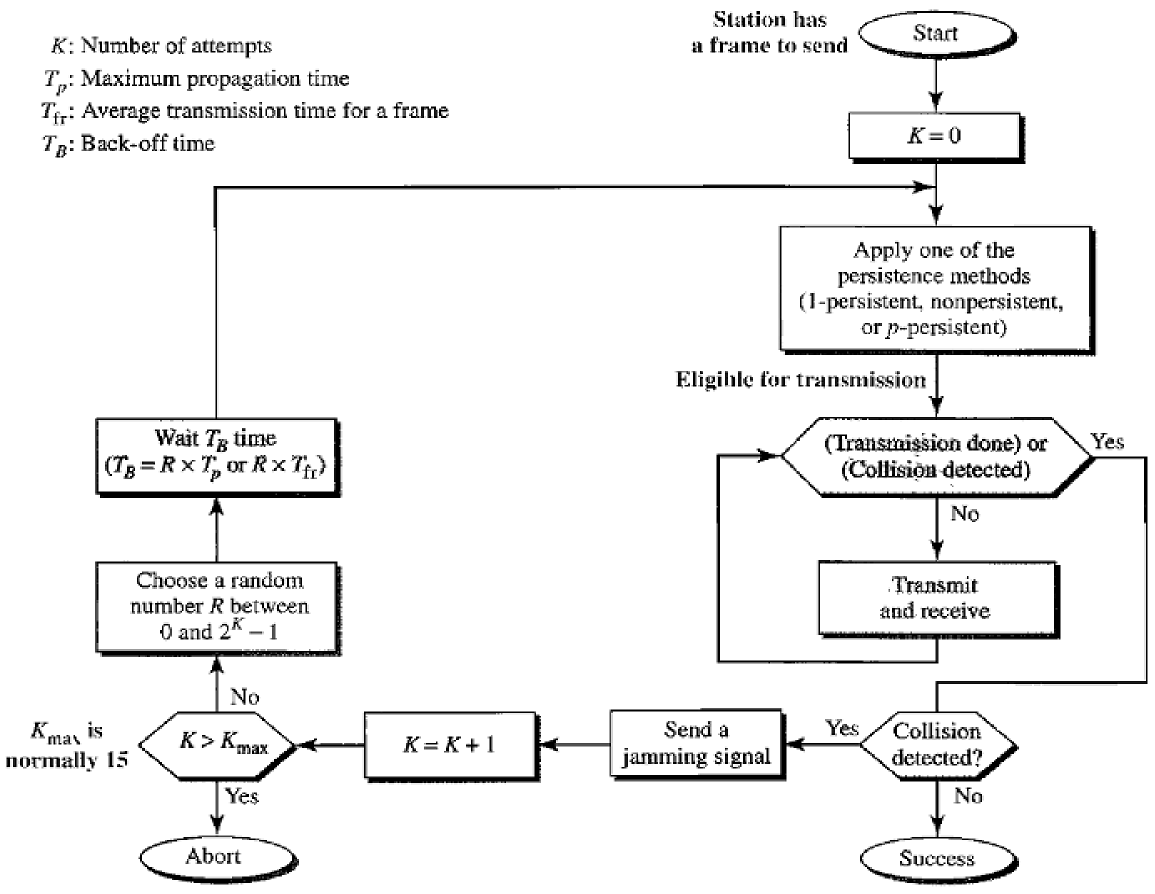
Minimum Frame Size (2Tp)

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.



K: Number of attempts
T_p: Maximum propagation time
T_fr: Average transmission time for a frame
T_B: Back-off time
Station has a frame to send
Start
K = 0
Apply one of the persistence methods (1-persistent, nonpersistent, or p-persistent)
Eligible for transmission
(Transmission done) or (Collision detected)
Yes
No
Transmit and receive
Collision detected?
Yes
No
Success
Send a jamming signal
K = K + 1
K > K_max
K_max is normally 15
Yes
Abort
No
Choose a random number R between 0 and 2^K − 1
Wait T_B time (T_B = R × T_p or R × T_fr)

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

Wireless networks cannot detect collisions, hence we need to avoid collision.

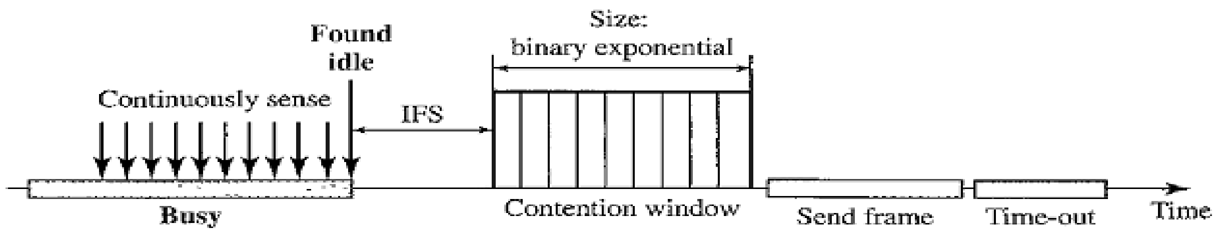
Need for CSMA/CA

- In a wired network, the received signal has almost the 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 and the receiver. The detected energy almost doubles when there is a collision.
- In a wireless network much of the **Sent Energy** is lost in transmission. The received signal has very little energy. Therefore a collision may add only 5 to 10 percent additional energy. This is not useful for effective collision detection.
- Hence in a wireless networks we need to avoid collisions instead of detecting collisions. In order to avoid collisions CSMA/CA was introduced for wireless networks.

CSMA/CA uses three strategies to avoid collisons:

- InterFrame Space
- Contention Window
- Acknowledgements

The below figure shows the timing in CSMA/CA:



Interframe Space (IFS)

- Collisions are avoided by deferring transmission even if the channel is found idle. When an idle channel is found, the station does not send immediately. It waits for a period of time called the interframe space or IFS.
- Even though the channel may appear idle when it is sensed, a distant station may have already started transmitting.
- The distant station's signal has not yet reached this station. The IFS time allows the front of the transmitted signal by the distant station to reach this station.
- If after the IFS time the channel is still idle, the station can send, but it still needs to wait a time equal to the contention time.
- The IFS variable can also be used to prioritize stations or frame types. For example, a station that is assigned a shorter IFS has a higher priority.

Contention Window

- The contention window is an amount of time divided into slots.
- A station that is ready to send chooses a random number of slots as its wait time.
- The number of slots in the window changes according to the binary exponential back-off strategy. This means that it is set to one slot the first time and then doubles each time the station cannot detect an idle channel after the IFS time.

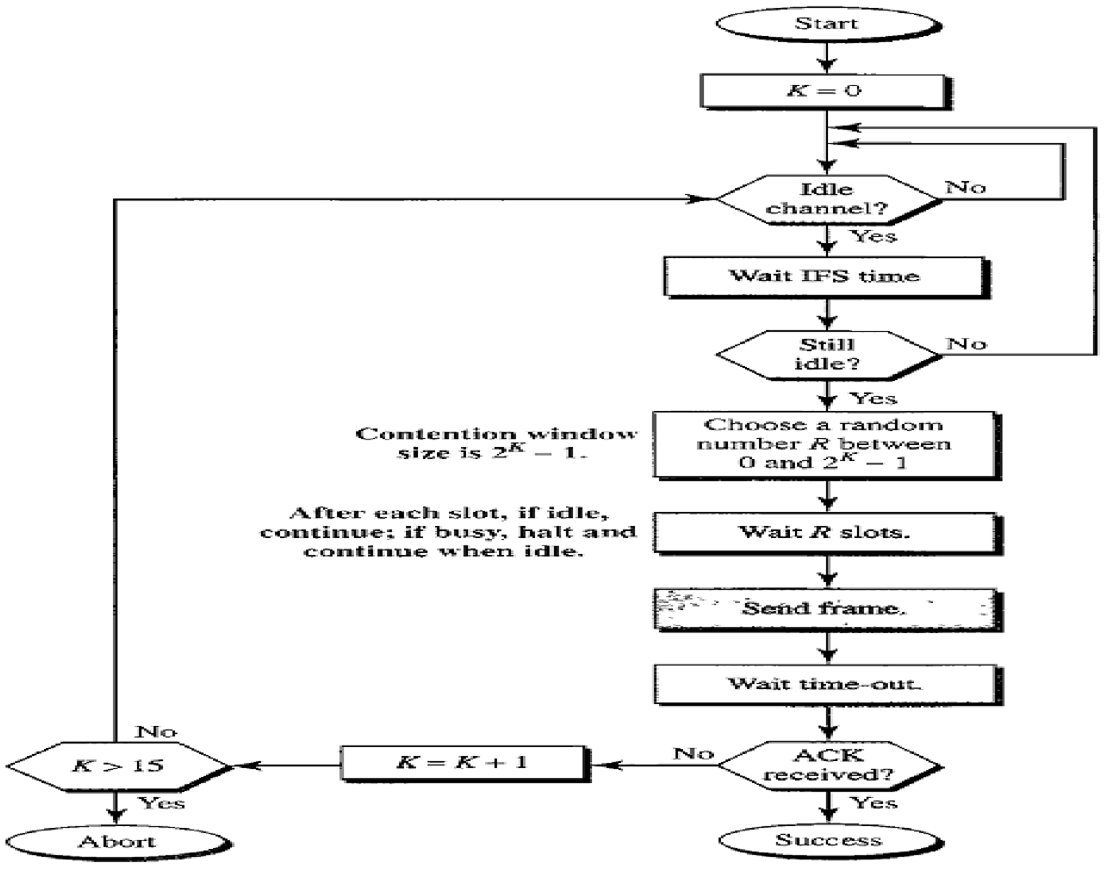
- A random outcome defines the number of slots taken by the waiting station.
- In contention window, the station needs to sense the channel after each time slot.
- In CSMAlCA, if the station finds the channel busy, it does not restart the timer of the contention window instead it stops the timer and restarts it when the channel becomes idle. This gives priority to the station with the longest waiting time.

Acknowledgment

- The positive acknowledgment and the time-out timer can help guarantee that the receiver has received the frame.
- Acknowledgement is needed even though we take all the precautions, there still may be a collision resulting in destroyed data and the data may be corrupted during the transmission.

Procedure

- The channel needs to be sensed before and after the IFS.
- The channel also needs to be sensed during the contention time.
- For each time slot of the contention window, the channel is sensed.
- If it is found idle, the timer continues.
- If the channel is found busy, the timer is stopped and continues after the timer becomes idle a-again.


Start
$K = 0$
Idle channel?
No
Yes
Wait IFS time
Still idle?
No
Yes
Contention window size is $2^K - 1$.
Choose a random number R between 0 and $2^K - 1$
After each slot, if idle, continue; if busy, halt and continue when idle.
Wait R slots.
Send frame.
Wait time-out.
No
$K > 15$
$K = K + 1$
No
ACK received?
Yes
Yes
Abort
Success

CONTROLLED ACCESS

In controlled access, the stations consult one another to find which station has the right to send. A station cannot send unless it has been authorized by other stations.

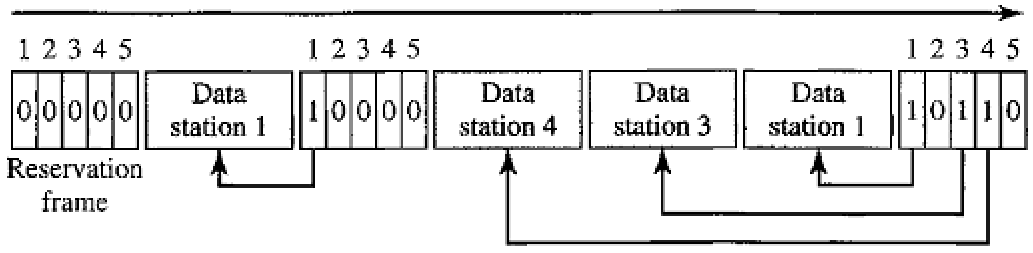
There are 3 methods in controlled access

1. Reservation
2. Polling
3. Token Passing

Reservation

In the reservation method, a station needs to make a reservation before sending data. Time is divided into intervals. In each interval, a reservation frame precedes the data frames sent in that interval.

- If there are *N* stations in the system, there are exactly *N* reservation mini-slots in the reservation frame. Each mini-slot belongs to a station.
- When a station needs to send a data frame, it makes a reservation in its own mini-slot.
- The stations that have made reservations can send their data frames after the reservation frame.
- The above figure shows a situation with 5-stations and a 5-minislot reservation frame.
- In the first interval, only stations 1, 3, and 4 have made reservations.
- In the second interval, only station 1 has made a reservation.



Polling

- Polling works with topologies in which one device is designated as a primary station and the other devices are secondary stations such as Star Topology.
- All data exchanges must be made through the primary device even when the ultimate destination is a secondary device.
- The primary device controls the link; the secondary devices follow its instructions.
- It is up to the primary device to determine which device is allowed to use the channel at a given time. Therefore, the primary device is always the initiator of a session.
- If the primary is neither sending nor receiving data, it knows the link is available.

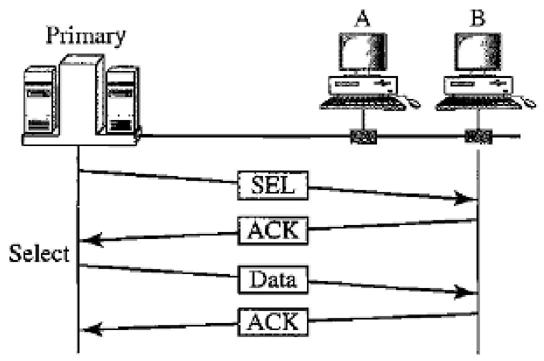
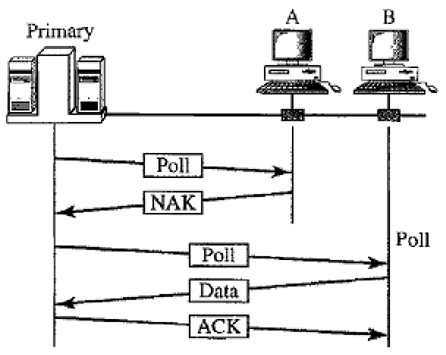
Polling uses two functions: Select and Poll functions

Select:

- If the primary station wants to send data, it tells the secondary station to get ready to receive the data.
- So the primary station must alert the secondary station to the upcoming transmission and wait for an acknowledgment of the secondary's ready status.
- Before sending data, the primary creates and transmits a select (SEL) frame, one field of select frame includes the address of the intended secondary station.

Poll:

- If the primary station wants to receive data, it asks the secondary stations if they have anything to send.
- When the primary station is ready to receive data, it must ask (poll) each device in turn if it has anything to send.
- When the poll frame is approached by one of the secondary stations it responds with either with a NAK frame if it has nothing to send or with Data frame if it has something to send.
- If the response received is NAK frame then the primary station polls the next secondary station until it finds the one station with the data to send.
- When the response is a data frame, the primary station reads the frame and returns an acknowledgment ACK frame to the respective secondary station.

Select function	Poll function
	 Primary A B Poll NAK Poll Poll Data ACK

Token Passing

In this method the stations in a network are organized in a logical ring. That means:

- For each station, there is a predecessor and a successor.
- The predecessor is the station which is logically before the station in the ring.
- The successor is the station which is after the station in the ring.
- The current station is the one that is accessing the channel now.
- The right to this access has been passed from the predecessor to the current station.
- The right will be passed to the successor when the current station has no more data to send.

Steps to get the right to access the channel passed from one station to another station

- A special packet called a token circulates through the ring.
- Possession of the token gives the station the right to access the channel and send its data.
- When a station has some data to send, it waits until it receives the token from its predecessor. It then holds the token and sends its data.
- When the station has no more data to send, it releases the token, passing it to the next logical station in the ring.
- The station cannot send data until it receives the token again in the next round.
- In this process, when a station receives the token and has no data to send, it just passes the data to the next station.

Token Management

- Stations must be limited in the time they can have possession of the token.
- The token must be monitored to ensure it has not been lost or destroyed.
- If a station that is holding the token fails, the token will disappear from the network.
- Another function of token management is to assign priorities to the stations and to the types of data being transmitted.
- Token management is needed to make low-priority stations release the token to high-priority stations.

Logical Ring

Physical Ring topology

- In the physical ring topology, when a station sends the token to its successor, the token cannot be seen by other stations; the successor is the next one in line.
- This means that the token does not have to have the address of the next successor.
- The problem with this topology is that if one of the links between two adjacent stations fails, the whole system fails.

Dual Ring Topology

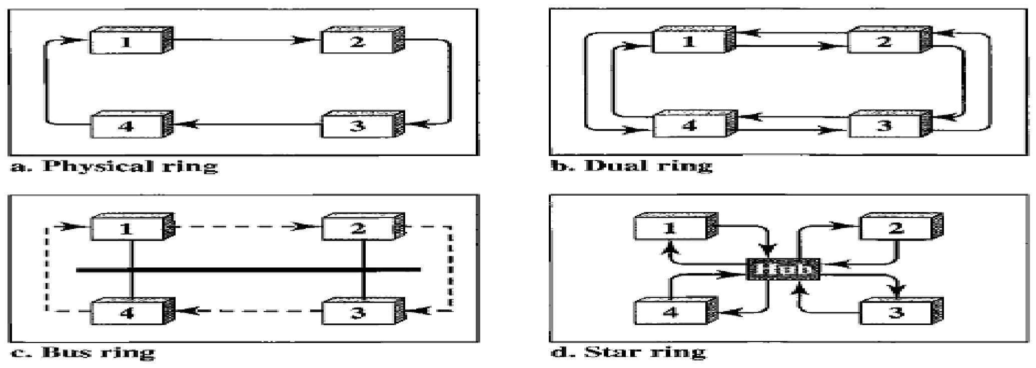
- The dual ring topology uses a second ring or auxiliary ring which operates in the reverse direction compared with the main ring.
- The second ring is for emergencies only.
- If one of the links in the main ring fails, the system automatically combines the two rings to form a temporary ring.
- After the failed link is restored, the auxiliary ring becomes idle again.
- FDDI and CDDI uses this topology.

Bus Ring Topology or Token Bus

- The stations are connected to a single cable called a bus.
- They make a logical ring, because each station knows the address of its successor and also predecessor.
- When a station has finished sending its data, it releases the token and inserts the address of its successor in the token.
- Only the station with the address matching the destination address of the token gets the token to access the shared media.
- The Token Bus LAN uses this topology.

Star Ring Topology

- The physical topology is a star topology.
- There is a hub that acts as the connector.
- The wiring inside the hub makes the ring and the stations are connected to this ring through the two wire connections.
- This topology makes the network less prone to failure because if a link goes down, it will be bypassed by the hub and the rest of the stations can operate.
- Also adding and removing stations from the ring is easier.
- This topology is still used in the Token Ring LAN designed by IBM.



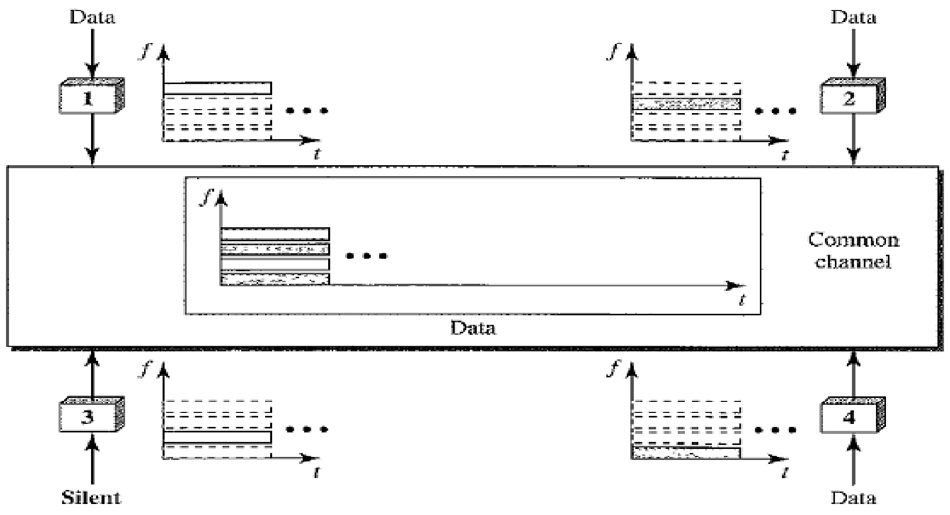
CHANNELIZATION

Channelization is a multiple-access method in which the available bandwidth of a link is shared in time, frequency, or through code, between different stations.
There are 3 channelization protocols:

1. FDMA
2. TDMA
3. CDMA

Frequency-Division Multiple Access (FDMA)

- In FDMA the available bandwidth is divided into frequency bands.
- Each station is allocated a frequency band to send its data. (i.e.) each frequency band is reserved for a specific station, and it belongs to the station all the time.
- Each station also uses a **band-pass filter** to confine the transmitter frequencies.
- Guard Bands are used to separate one station from other station to prevent the interferences between the stations.
- FDMA specifies a predetermined frequency band for the entire period of communication. Hence the data streaming can easily be used with FDMA.

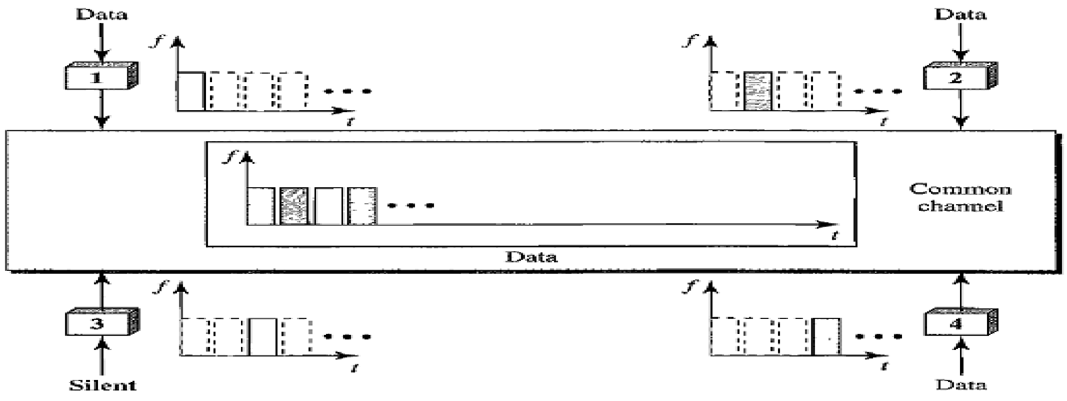


Time-Division Multiple Access (TDMA)

In TDMA the stations share the bandwidth of the channel in time. Each station is allocated a time slot during which it can send data.

- To achieve synchronization between the stations it introduces the synchronization bits at the beginning of each slot.

- So, each station needs to know the beginning of its time slot and the location of its time slot.



- This may be difficult because of propagation delays introduced in the system if the stations are spread over a large area. To compensate for the delays, we can insert **Guard Times**.

<u>Code-Division Multiple Access (CDMA)</u>

- In CDMA, one channel carries all transmissions simultaneously.
- CDMA simply means communication with different codes
- CDMA differs from FDMA because only one channel occupies the entire bandwidth of the link.
- CDMA differs from TDMA because all stations can send data simultaneously and there is no timesharing.

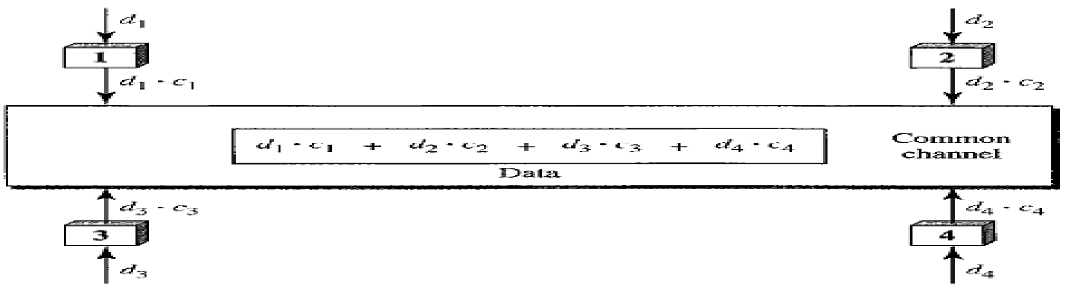
Example:
From the below figure, there are four stations 1, 2, 3 and 4 are connected to the same channel.

- d1, d2, d3, d4 are the data from stations 1,2,3,4 respectively.
- c1, c2, c3, c4 are the codes assigned to stations 1,2,3,4 respectively.
- (d1.c1), (d2.c2), (d3.c3),(d4.c4) are the multiplication of data and codes of stations 1,2,3,4 respectively. The sum of all these terms is equal to the data.

data=(d1.c1)+(d2.c2)+(d3.c3)+(d4.c4)

There are two conditions:

1. If we multiply each code by another, we get 0. (c1.c2=0, c1.c3=0, c2.c4=0 etc)
2. If we multiply each code by itself, we get 4 (c1.c1=4, c2.c2=4, c3.c3=4, c4.c4=4).



Any station that wants to receive data from one of the other three multiplies the data on the channel by the code of the sender.

For example, suppose stations 1 and 2 are talking to each other.

- Station 2 wants to hear what station 1 is saying. It multiplies the data on the channel by the code c1 of station 1.
- From the above conditions, (c1 . c1) is 4, but (c2 . c1), (c3 . c1), and (c4 . c1) are all 0's, station 2 divides the result by 4 to get the data from station 1.

data =(d1.c1 + d2.c2 + d3.c3 + d4.c4).c1

data =d1.c1.c1 + d2.c2.c1 + d3.c3.c1 + d4.c4.c1

data=4.d1

WIRED LANS: ETHERNET (802.3)

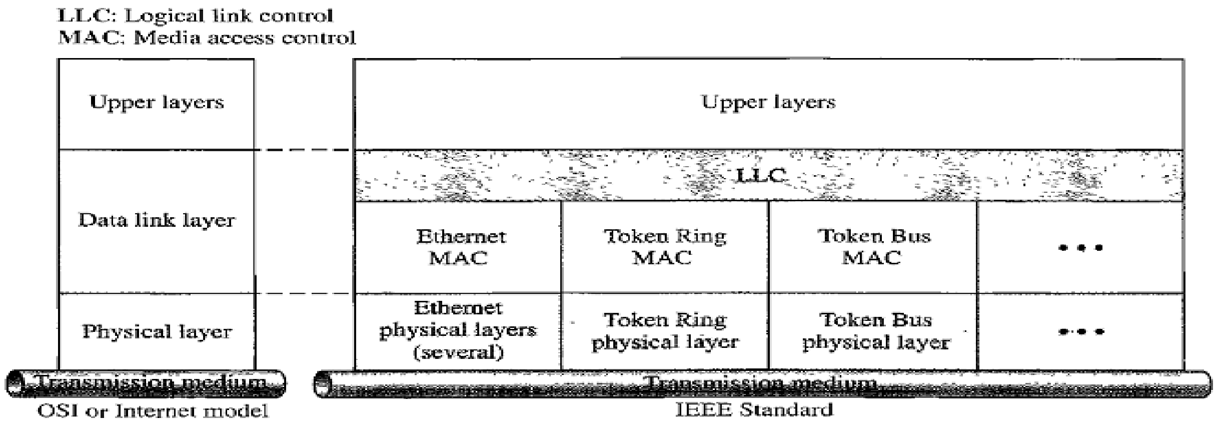
Local Area Network (LAN) is a computer network that is designed for limited geographic area such as building or campus. LAN is a shared resource.

LAN technologies: Ethernet, Token Ring, Token Bus, FDD, ATM LAN.

IEEE STANDARDS for LAN

The IEEE has subdivided the data link layer into two sublayers:

1. Logical Link Control (LLC)
2. Media Access Control (MAC)

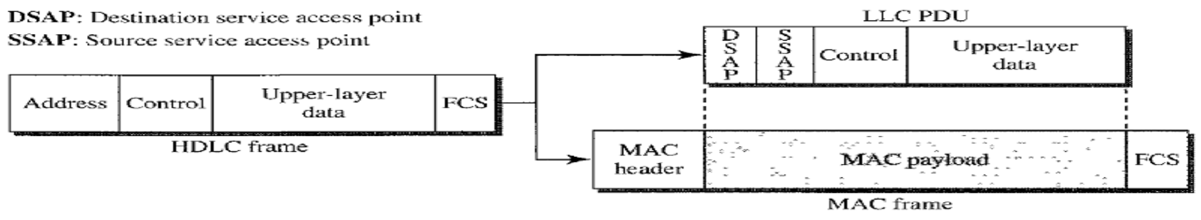


Logical Link Control (LLC)

- In IEEE Project 802, the logical link layer takes the responsibilities of flow control, error control, and part of the framing duties.
- Framing is handled in both the LLC sublayer and the MAC sublayer.
- The LLC provides one single data link control protocol for all IEEE LANs.
- The LLC is different from the media access control sublayer, which provides different protocols for different LANs.
- A single LLC protocol can provide interconnectivity between different LANs because it makes the MAC sublayer transparent.

Framing

- LLC defines a protocol data unit (PDU).
- The Header contains DSAP,SSAP, Control field, upper layer data field.
- Control field is used for flow and error control.
- **Destination Service Access Point** (DSAP) and the **Source Service Access Point** (SSAP) fields define the upper-layer protocol at the source and destination that uses LLC.



Note: A frame defined in HDLC is divided into a PDU at the LLC sublayer and a frame at the MAC sub-layer

Media Access Control (MAC)

- IEEE Project 802 has created a sublayer called **Media Access Control** that defines the specific access method for each LAN.
- For example, it defines *CSMA/CD* as the media access method for Ethernet LANs and the token-passing method for Token Ring and Token Bus LANs.

The original Ethernet was created in 1976 at Xerox's Palo Alto Research Center (PARC). Since then, it has gone through four generations:

- Standard Ethernet (l Mbps)
- Fast Ethernet (100 Mbps)
- Gigabit Ethernet (l Gbps)
- Ten-Gigabit Ethernet (l0 Gbps)

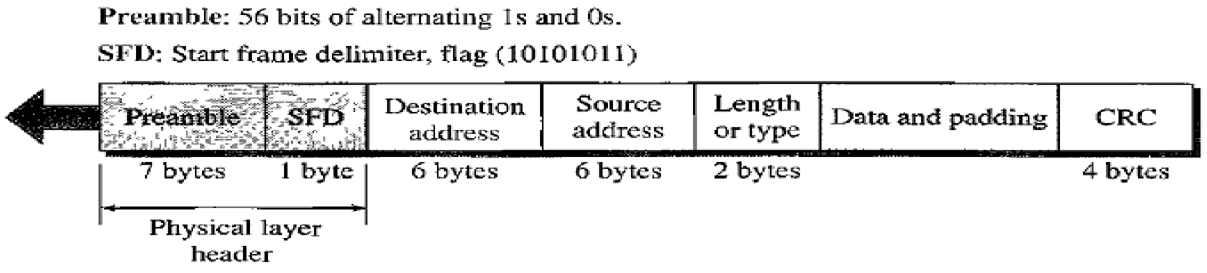
STANDARD ETHERNET (IEEE 802.3)

MAC Sub-layer

In Standard Ethernet, the MAC sublayer governs the operation of the access method. It also frames data received from the upper layer and passes them to the physical layer.

Frame Format

The Ethernet frame contains **seven** fields: Preamble, SFD, DA, SA, Length or Type of protocol data unit (PDU), Upper-layer data, and CRC.



Preamble

- The first field of the 802.3 frame contains 7 bytes (56 bits) of alternating 0's and 1's that alerts the receiving system to the coming frame and enables it to synchronize its input timing.
- The pattern provides only an alert and a timing pulse.
- The 56-bit pattern allows the stations to miss some bits at the beginning of the frame.

Note: The preamble is actually added at the physical layer and is not (formally) part of the frame.

Start Frame Delimiter (SFD)

- The second field (l byte: 10101011) signals the beginning of the frame.
- The SFD warns the stations that "This is the last chance for synchronization".
- The last 2 bits is 11 and alerts the receiver that the next field is the destination address.

Destination address (DA)

- The DA field is 6 bytes and contains the physical address of the destination station (i.e.) stations to receive the packet.

Source address (SA)

- The SA field is also 6 bytes and contains the physical address of the sender of the packet.

Length or Type

- This field is defined as a type field or length field. Both uses are common today.
- The original Ethernet used this field as the **Type field** to define the upper-layer protocol using the MAC frame.
- The IEEE standard used it as the length field to define the number of bytes in the data field.

Data

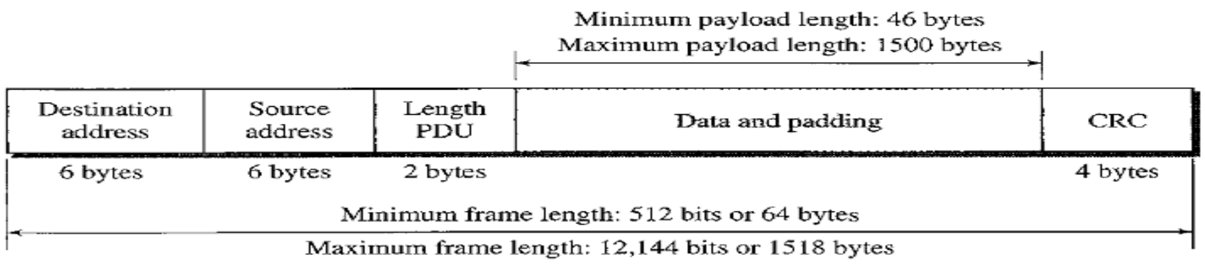
- This field carries data encapsulated from the upper-layer protocols.
- It is a minimum of 46 bytes and a maximum of 1500 bytes.

CRC

- The last field contains Error Detection information. The Length of this field is 4 byte (32-bit) hence a CRC-32 is used.

<u>**Frame Length**</u>

Ethernet has imposed restrictions on both the minimum and maximum lengths of a frame.


Minimum payload length: 46 bytes
Maximum payload length: 1500 bytes
Destination address
Source address
Length PDU
Data and padding
CRC
6 bytes
6 bytes
2 bytes
4 bytes
Minimum frame length: 512 bits or 64 bytes
Maximum frame length: 12,144 bits or 1518 bytes

Minimum Length (64 bytes=512 bits)

- The minimum length restriction is required for the correct operation of CSMA/CD.
- Out of 64 bits of frame 18 bits contains Header(14 btis) and Trailer(4 bits).
- The header contains DA (6 bytes), SA(6 bytes), Length or Type (2 bytes).
- The trailer contains CRC (4 bytes).
- Data and Padding field is a 46 bytes field.
- If the upper-layer packet is less than 46 bytes, padding is added to make up the difference.

Maximum Length (1518 bytes=12,144 bits)

- The IEEE standard 802.3 defines the maximum length of a frame as 1518 bytes (without preamble and SFD field).
- If we subtract the 18 bytes of header and trailer, the maximum length of the payload is 1500 bytes (1518-18=1500 bytes).

Note: The maximum length restriction has two historical reasons.

1. Memory was very expensive when Ethernet was designed: a maximum length restriction helped to reduce the size of the buffer.
2. The maximum length restriction prevents one station from monopolizing the shared medium, blocking other stations that have data to send.

Addressing

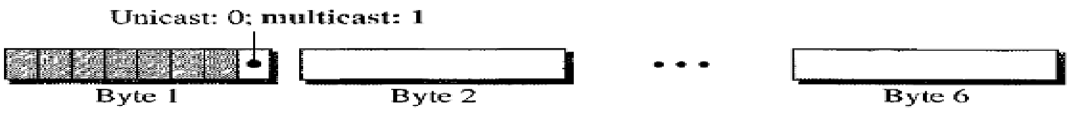
- Each station on an Ethernet network (such as a PC, workstation, or printer) has its own network interface card (NIC).
- The NIC fits inside the station and provides the station with a 6-byte physical address.
- The Ethernet address is 6 bytes (48 bits), normally written in hexadecimal notation, with a colon between the bytes.

Example: **06 : 01 : 02 : 01 : 2C : 4B**

There are 3 types addressing:

1. Unicast Addressing
2. Multicast Addressing
3. Broadcast Addressing

- A source address is always a unicast address-the frame comes from only one station.
- The destination address can be unicast, multicast, or broadcast.
- If the least significant bit of the first byte in a destination address is 0, the address is **Unicast Address,** otherwise (i.e. 1) it is **Multicast**.



Unicast Destination Address defines only one recipient; the relationship between the sender and the receiver is one-to-one.
Multicast Destination Address defines a group of addresses; the relationship between the sender and the receivers is one-to-many.
Broadcast Address is a special case of the multicast address; the recipients are all the stations on the LAN. A broadcast destination address is 48-1's (all 1's in the address).

Example:

Unicast Address	4**A**:30:10:21:1O:1A	A in binary is 1010 (even)
Multicast Address	4**7**:20:1B:2E:08:EE	7 in binary is 0111 (odd)
Broadcast Address	**FF:FF:FF:FF:FF:FF**	all digits are F's

Address format that is sent on transmission line:

- The transmission is left-to-right and byte by byte.
- For each byte, the least significant bit is sent first and the most significant bit is sent last.
- This means that the bit that defines an address as unicast or multicast arrives first at the receiver.

Let us take the address **11100010 00000100 11011000 01110100 00010000 01110111**
In the above bit-stream, the left most byte 11100010 will be taken as the first byte and right most bit 0 will be taken as the first bit.

Hence above bitstream can be transmitted on to the transmission line is:

Original	**11100010 00000100 11011000 01110100 00010000 01110111**
On Link	**01000111 00100000 00011011 00101110 00001000 11101110**

Access Method: CSMA/CD

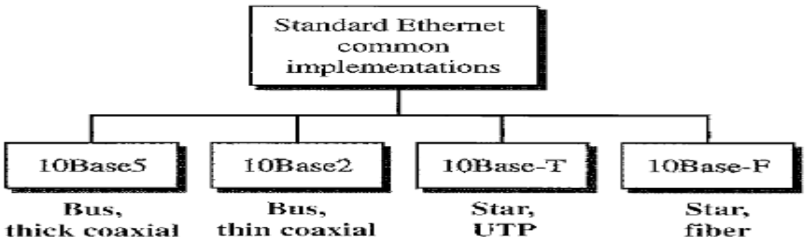
Standard Ethernet uses 1-persistent CSMA/CD.

Slot Time

Slot time = round-trip time + time required to send the jam sequence

- In an Ethernet network, the round-trip time required for a frame to travel from one end of a maximum-length network to the other plus the time needed to send the jam sequence is called the slot time.
- The slot time in Ethernet is defined in bits.
- It is the time required for a station to send 512 bits.
- This means that the actual slot time depends on the data rate; for traditional 10-Mbps Ethernet it is 51.2µs.

4 Categories of Standard Ethernet



10Base5: Thick Ethernet

- 10Base5 is also called Thick Ethernet, or Thicknet. It is too stiff to bend .
- 10Base5 was the first Ethernet specification to use a bus topology with an external **transceiver** (transmitter/receiver) connected via a tap to a thick coaxial cable.
- The transceiver is responsible for transmitting, receiving, and detecting collisions.
- The maximum length of the coaxial cable must not exceed *500* meters.
- If the length exceeds more than 500m there is an excessive degradation of signal.
- The transceiver is connected to the station via a transceiver cable that provides separate paths for sending and receiving. This means that collision can only happen in the coaxial cable.

10Base2: Thin Ethernet or Cheapernet

- It uses a bus topology and the cable is much thinner and more flexible. The cable can be bent to pass very close to the stations.
- The length of each segment cannot exceed *185* m (~= 200 m) due to the high level of attenuation in thin coaxial cable.

- The transceiver is normally part of the network interface card (NIC), which is installed inside the station.
- The collision here occurs in the thin coaxial cable.

Advantages:

- Thin coaxial cable is less expensive than thick coaxial cable.
- Thin coaxial cable is very flexible hence installation is simpler.

10Base-T: Twisted-Pair Ethernet

- 10Base-T uses a physical star topology.
- The stations are connected to a hub via two pairs of twisted cable.
- Two pairs of twisted cable create two paths one for sending and one for receiving between the station and the hub.
- Any collision here happens in the hub.
- The maximum length of the twisted cable here is defined as 100 m, to minimize the effect of attenuation in the twisted cable.

10Base-F: Fiber Ethernet

- 10Base-F uses a star topology to connect stations to a hub.
- The stations are connected to the hub using two fiber-optic cables.
- The maximum length of the fiber optic cable is 2000m.

Characteristics	**10Base5**	**10Base-2**	**10Base-T**	**10Base-F**
Media	Thick Coaxial Cable	Thin Coaxial Cable	2UTP	2Fiber
Max.Length	500m	200m	100m	2000m
Line Encoding	Manchester	Manchester	Manchester	Manchester
Collision	Coaxial Cable	Coaxial Cable	Hub	Hub

Physical Layer

The Standard Ethernet defines several physical layer implementations.

Encoding and Decoding

- All standard implementations use digital signaling (baseband) at 10 Mbps.
- At the sender, data are converted to a digital signal using the Manchester scheme.
- At the receiver, the received signal is interpreted as Manchester and decoded into data.
- Manchester encoding is self-synchronous, providing a transition at each bit interval.

Wireless LAN (802.11)

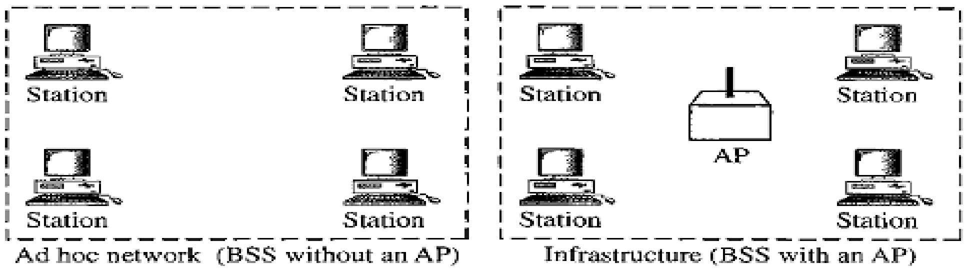
Wireless communication means connecting devices without the use of cables.

Architecture of IEEE 802.11

The standard defines two kinds of services:
1. Basic Service Set (BSS)
2. Extended Service Set (ESS)

Basic Service Set (BSS)

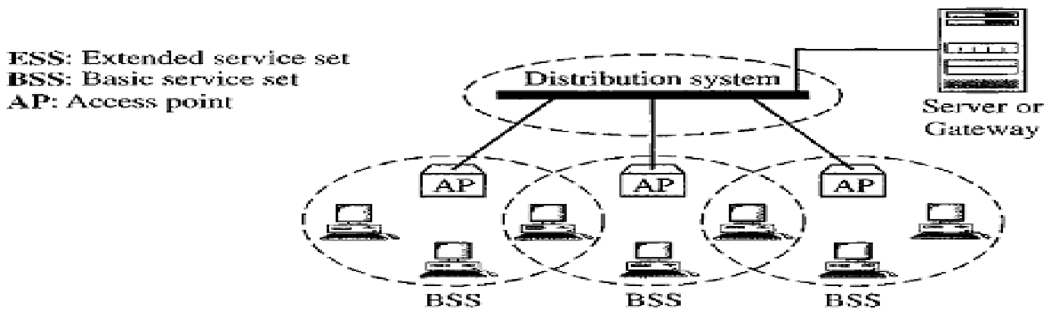
- IEEE 802.11 defines the basic service set (BSS) as the building block of a wireless LAN.
- BSS is made of stations or **Mobile Wireless Stations** and the **Access Point** (AP).
- Access Point is an optional central base station.
- **Ad-hoc Architecture:** The BSS without an Access Point is a stand-alone network and cannot send data to other BSSs. It is called an Ad-hoc Architecture. In this architecture, stations can form a network without the need of an Access Point; they can locate one another and agree to be part of a BSS.
- A BSS with an AP is sometimes referred to as an **Infrastructure Network**.


Station
Station
Station
Station
Ad hoc network (BSS without an AP)
Station
Station
AP
Station
Station
Infrastructure (BSS with an AP)

Extended Service Set (ESS)

- ESS is made up of two or more BSSs with Access Points.
- In this case, the BSSs are connected through a *distribution system,* which is usually a wired LAN. The distribution system connects the Access Points in the BSS's.
- The extended service set uses two types of stations: mobile and stationary.
- The mobile stations are normal stations inside a BSS.
- The stationary stations are Access Point stations that are part of a wired LAN.
- When BSSs are connected, the stations within reach of one another can communicate without the use of an AP.
- Communication between two stations in two different BSS usually occurs via two Access Points.

Note: A mobile station can belong to more than one BSS at the same time.



Station Types

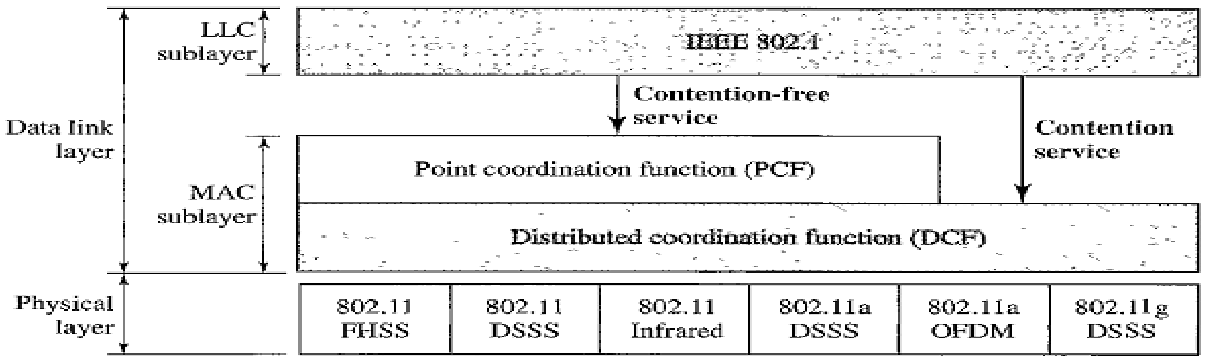
IEEE 802.11 defines three types of stations based on their mobility in a wireless LAN:

- **No-transition Station** A station with no-transition mobility is either stationary (not moving) or moving only inside a BSS.
- **BSS-transition Station** A station with BSS-transition mobility can move from one BSS to another, but the movement is confined inside one ESS.
- **ESS-transition Station** A station with ESS-transition mobility can move from one ESS to another.

MAC Sub-layer

IEEE 802.11 defines two MAC sub-layers:

1. Distributed Coordination Function (DCF)
2. Point Coordination Function (PCF)

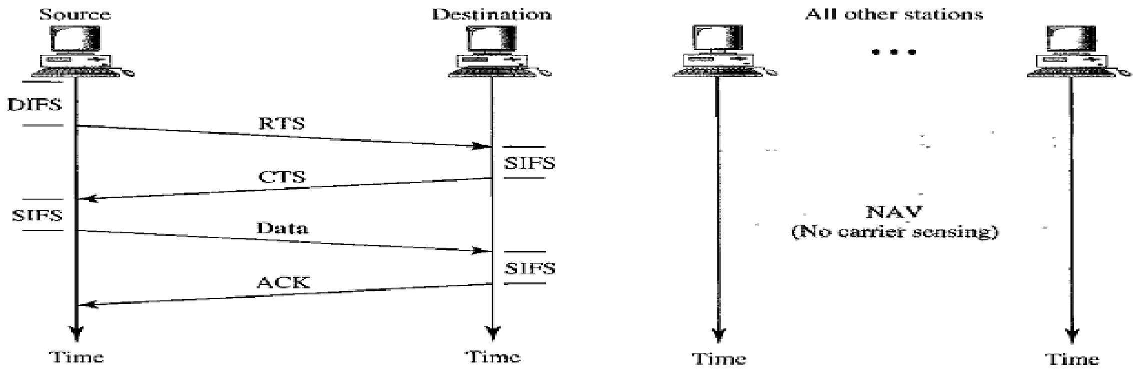


Wireless LANs cannot implement CSMA/CD for three reasons:

i. For collision detection a station must be able to send data and receive collision signals at the same time. This approach makes stations costly and increase bandwidth requirements.
ii. Collision may not be detected because of the hidden station problem.
iii. The distance between stations can be great. Signal fading could prevent a station at one end from hearing a collision at the other end.

Distributed Coordination Function (DCF)

DCF uses **CSMA/CA** as the access method. Process Flow Chart of CSMA/CA given below



1. Before sending a frame, the source station senses the medium by checking the energy level at the carrier frequency.
    i. The channel uses a persistence strategy with back-off until the channel is idle.

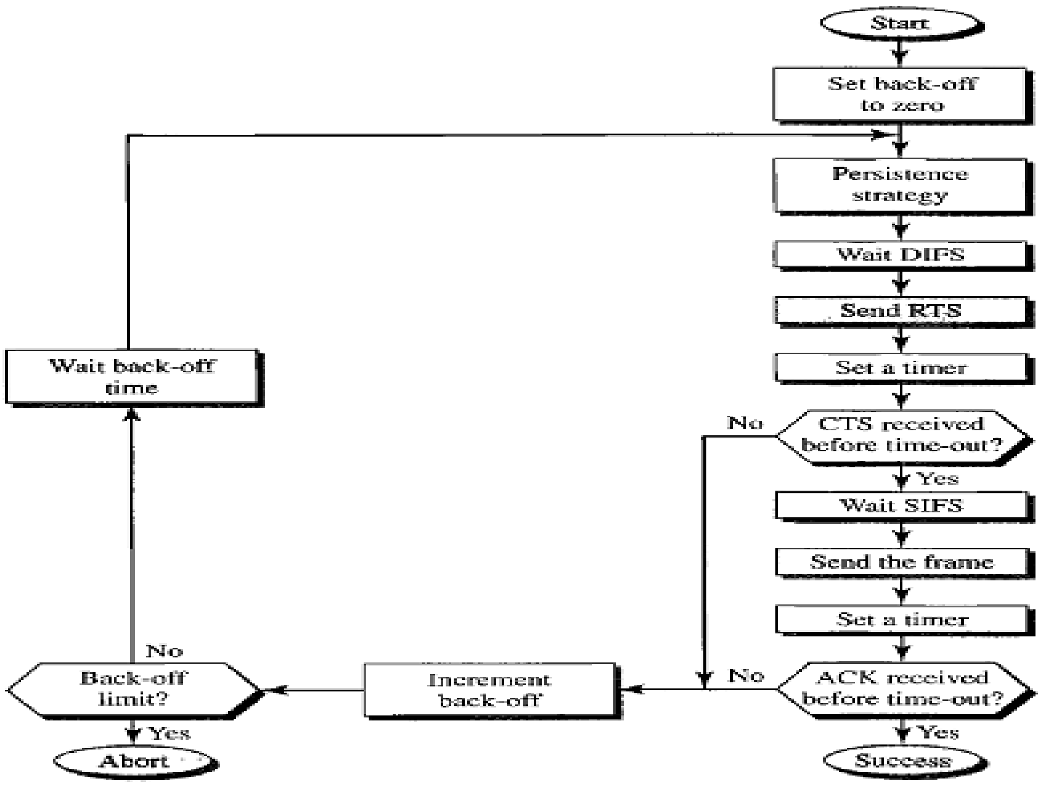
ii. After the station is found to be idle, the station waits for a period of time called the Distributed Inter-Frame Space (DIFS); then the station sends a control frame called the Request To Send (RTS).

2. After receiving the RTS and waiting a period of time called the Short Inter-Frame Space (SIFS), the destination station sends a control frame called the Clear To Send (CTS) to the source station. This control frame indicates that the destination station is ready to receive data.
3. The source station sends data after waiting an amount of time equal to SIFS.
4. The destination station waits for an amount of time equal to SIFS then sends an acknowledgment to source station to show that the frame has been received.

Note: Acknowledgment is needed in this protocol because the station does not have any mechanism to check for the successful arrival of its data at the destination.

Network Allocation Vector (NAV)

- NAV is a timer used to avoid the collision.
- When a station sends an RTS frame, it includes the duration of time that it needs to occupy the channel.
- The stations that are affected by this transmission create a timer called a **Network Allocation Vector** (NAV) that shows how much time must pass before these stations are allowed to check the channel for idleness.
- Each time a station accesses the system and sends an RTS frame, other stations start their NAV.

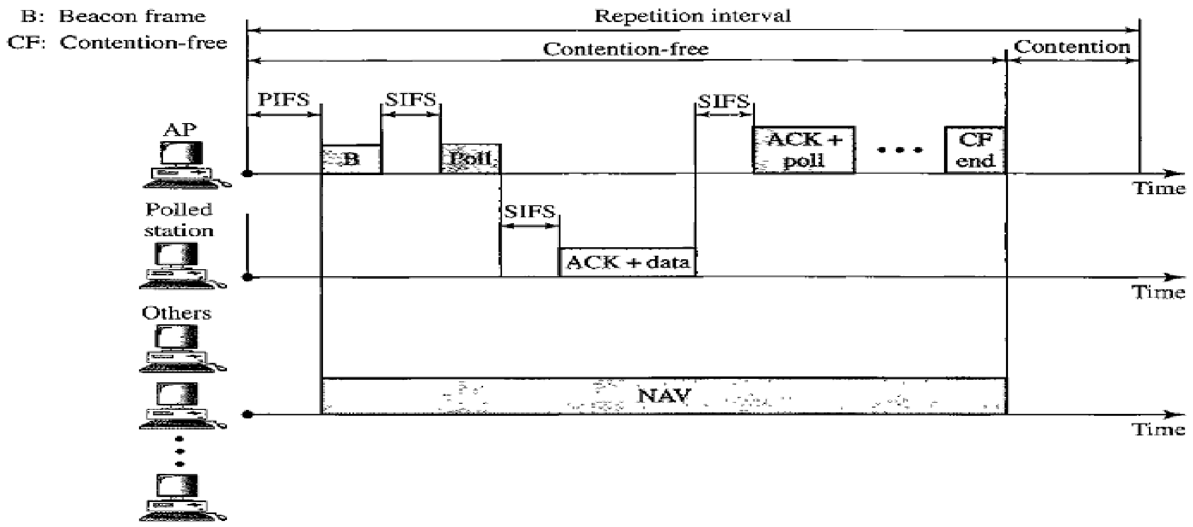


Collision during Handshaking

- Handshaking period is the time taken for the transition of RTS and CTS control frames.
- Two or more stations may try to send RTS frames at the same time. These control frames may collide.
- There is no mechanism for collision detection, the sender assumes there has been a collision if it has not received a CTS frame from the receiver.
- The back-off strategy is employed, and the sender tries again.

Point Coordination Function (PCF)

- The PCF is an optional access method that can be implemented in an infrastructure network (i.e. BSS with Access Point). It is used mostly for time-sensitive transmission.
- PCF has a centralized, contention-free polling access method.
- The Access Point performs polling for stations that are capable of being polled.
- The stations are polled one station after another station if they have any data to send to the Access Point.
- To give priority to PCF over DCF, another set of Inter-Fame spaces has been defined: PIFS (PCF IFS) and SIFS.
- The SIFS is the same as that in DCF, but the PIFS is shorter than the DIFS (i.e.) at the same time, a station wants to use only DCF and an Access Point wants to use PCF then the priority is given to Access Point.



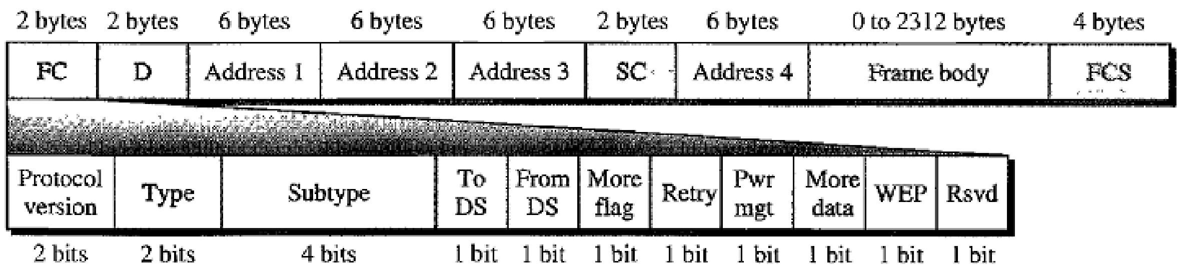
- Due to the priority of PCF over DCF, stations that only use DCF may not gain access to the medium.
- To prevent this, a repetition interval has been designed to cover both contention-free traffic (PCF) and contention-based traffic (DCF).
- The repetition interval, which is repeated continuously, starts with a special control frame, called a **Beacon Frame.**
- When the stations hear the beacon frame, they start their NAV for the duration of the contention-free period of the repetition interval.
- During the repetition interval, the PC (point controller) can send a poll frame, receive data, send an ACK, receives an ACK etc.

Fragmentation

- The wireless environment is very noisy and a corrupt frame has to be retransmitted. The protocol recommends fragmentation.
- Fragmentation is the division of a large frame into smaller ones. It is more efficient to resend a small frame than a large one.

Frame Format

The MAC layer frame consists of nine fields :



Frame control (FC) The FC field is 2 bytes long and defines the type of frame and some control information.

Table 14.1 *Subfields in FC field*

Field	*Explanation*
Version	Current version is 0
Type	Type of information: management (00), control (01), or data (10)
Subtype	Subtype of each type (see Table 14.2)
To DS	Defined later
From DS	Defined later
More flag	When set to 1, means more fragments
Retry	When set to 1, means retransmitted frame
Pwr mgt	When set to 1, means station is in power management mode
More data	When set to 1, means station has more data to send
WEP	Wired equivalent privacy (encryption implemented)
Rsvd	Reserved

D: In all frame types except one, this field defines the duration of the transmission that is used to set the value of NAV. In one control frame, this field defines the ID of the frame.

Addresses: There are four address fields, each 6 bytes long. The meaning of each address field depends on the value of the *To* DS and *From* DS subfields.

Sequence Control: This field defines the sequence number of the frame to be used in flow control.

Frame Body: This field, which can be between 0 and 2312 bytes, contains information based on the type and the subtype defined in the FC field.

FCS: The FCS field is 4 bytes long and contains a CRC-32 error detection sequence.

<u>Frame Types</u>

A wireless LAN defined by IEEE 802.11 has three categories of frames:

i. **Management Frames (00)**
These frames are used for the initial communication between stations and access points.

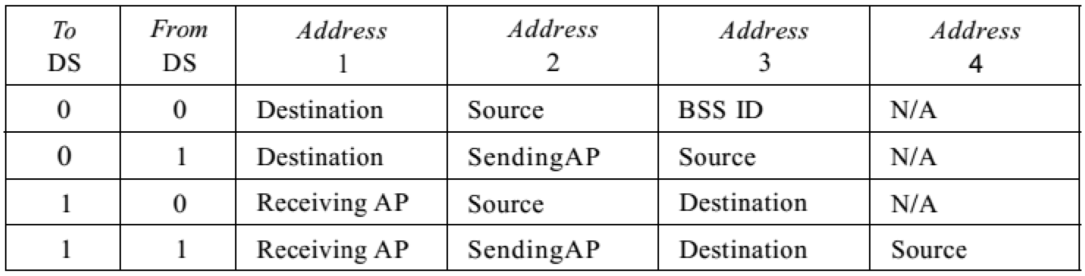
ii. Control Frames (01)
Control frames are used for accessing the channel and acknowledging frames.
The values of the subtype fields for frames:

Subtype	Meaning
1011	Request To Send (RTS)
1100	Clear To Send (CTS)
1101	Acknowledgement (ACK)

iii. Data Frames (10)
Data frames are used for carrying data and control information.

<u>Addressing Mechanism</u>

Flag fields such To DS, From DS can be either 0 or 1, resulting in four different situations. The interpretation of the four addresses (address 1 to address 4) in the MAC frame depends on the value of these flags.



To DS	*From* DS	*Address* 1	*Address* 2	*Address* 3	*Address* 4
0	0	Destination	Source	BSS ID	N/A
0	1	Destination	SendingAP	Source	N/A
1	0	Receiving AP	Source	Destination	N/A
1	1	Receiving AP	SendingAP	Destination	Source

Case 1: 00 (*To DS* = 0 and *From DS* = 0)

- The frame is not going to a distribution system *(To DS* = 0) and is not coming from a distribution system *(From DS* =0).
- The frame is going from one station in a BSS to another without passing through the distribution system.
- The ACK frame should be sent to the original sender.

Case 2: 01 (*To DS* = 0 and *From DS* = 1)

- The frame is coming from a distribution system *(From* DS = 1).
- The frame is coming from an Access Point and going to a station.
- The ACK should be sent to the Access Point.

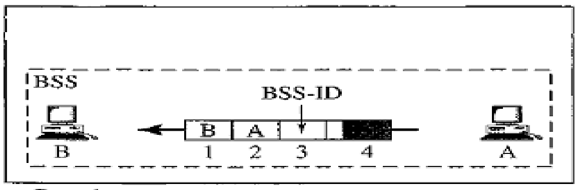
Case 3: 10 (*To DS* = 1 and *From DS* = 0)

- The frame is going to a distribution system *(To DS* = 1).
- The frame is going from a station to an Access Point.
- The ACK is sent to the original station.

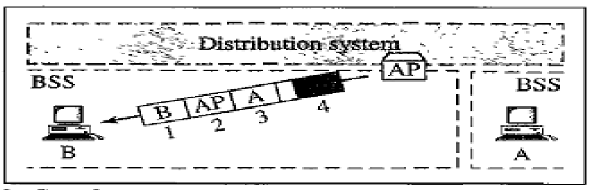
Case 4:11 (*To DS* =1 and *From DS* = 1)

- This is the case in which the distribution system is also wireless.
- The frame is going from one AP to another AP in a wireless distribution system.

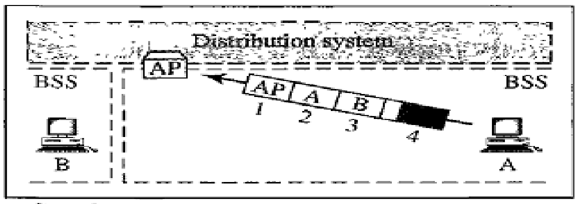
- We do not need to define addresses if the distribution system is a wired LAN because the frame in these cases has the format of a wired LAN frame (Ethernet, for example).
- Here, we need four addresses to define the original sender, the final destination, and two intermediate APs.



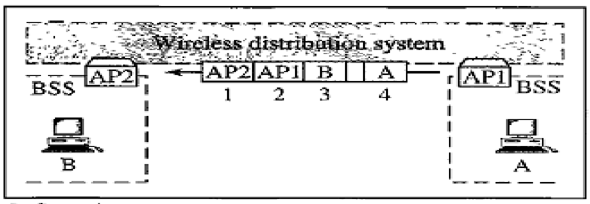
a. Case 1



b. Case 2



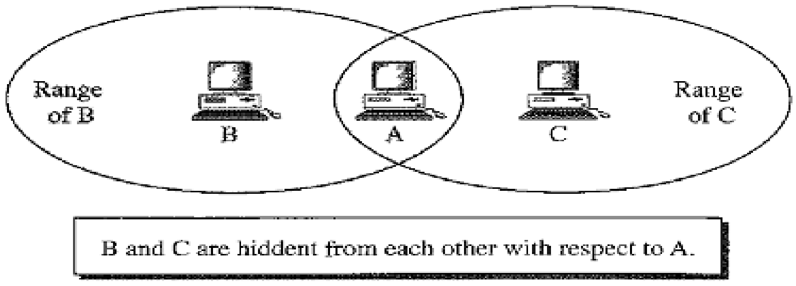
c. Case 3



d. Case 4

Hidden Station Problems

Consider the below figure for Hidden Station problem.



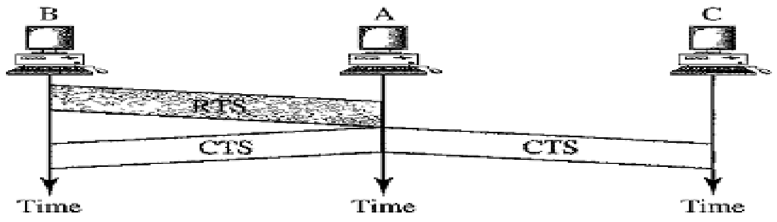
- Station B has a transmission range shown by the left oval, every station in this range can hear any signal transmitted by station B.
- Station C has a transmission range shown by the right oval , every station located in this range can hear any signal transmitted by C.
- Station C is outside the transmission range of B and station B is outside the transmission range of C.
- Station A is in the area covered by both B and C; Hence A can hear any signal transmitted by B or C.

Problem

- Station B is sending data to station A. In the middle of this transmission, station C also has data to send to station A.
- Station C is out of B's range and transmissions from B cannot reach C. Therefore C thinks the medium is free.
- Station C sends its data to A, which results in a collision at A because this station is receiving data from both B and C.
- In this case stations B and C are hidden from each other with respect to A.
- Hidden stations can reduce the capacity of the network because of the possibility of collision.

Solution

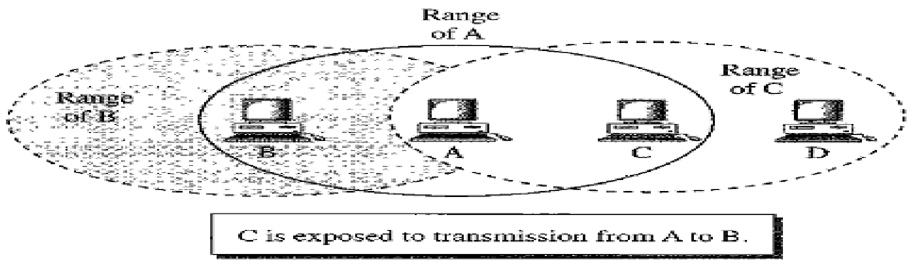
We avoid the above problem by using the **Handshake Frames** such as **RTS** & **CTS**.
Consider the below figure:



- The RTS message from B reaches A, but not C, because both B and C are within the range of A, the CTS message, which contains the duration of data transmission from B to A reaches C.
- Station C knows that some hidden station is using the channel and refrains from transmitting until that duration is over.

__**Exposed Station Problem**__

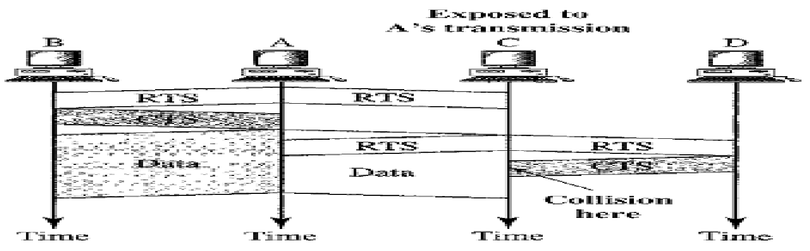
In this problem a station refrains from using a channel when it is available.



- Station A is transmitting to station B. Station C has some data to send to station D, which can be sent without interfering with the transmission from A to B.
- Station C is exposed to transmission from A; it hears what A is sending and thus refrains from sending. That means C is too conservative and wastes the capacity of the channel.

Solution

The Handshaking messages RTS and CTS cannot help in this case.



- Station C hears the RTS from A, but does not hear the CTS from B.
- Station C, after hearing the RTS from A, can wait for a time so that the CTS from B reaches A; it then sends an RTS to D to show that it needs to communicate with D.
- Both stations B and A may hear this RTS, but station A is in the sending state, not the receiving state. Station B responds with a CTS.

The problem is here.

If station A has started sending its data, station C cannot hear the CTS from station D because of the collision; it cannot send its data to D. It remains exposed until A finishes sending its data