UNIT II

DATA LINK LAYER

Contents

Data link Layer

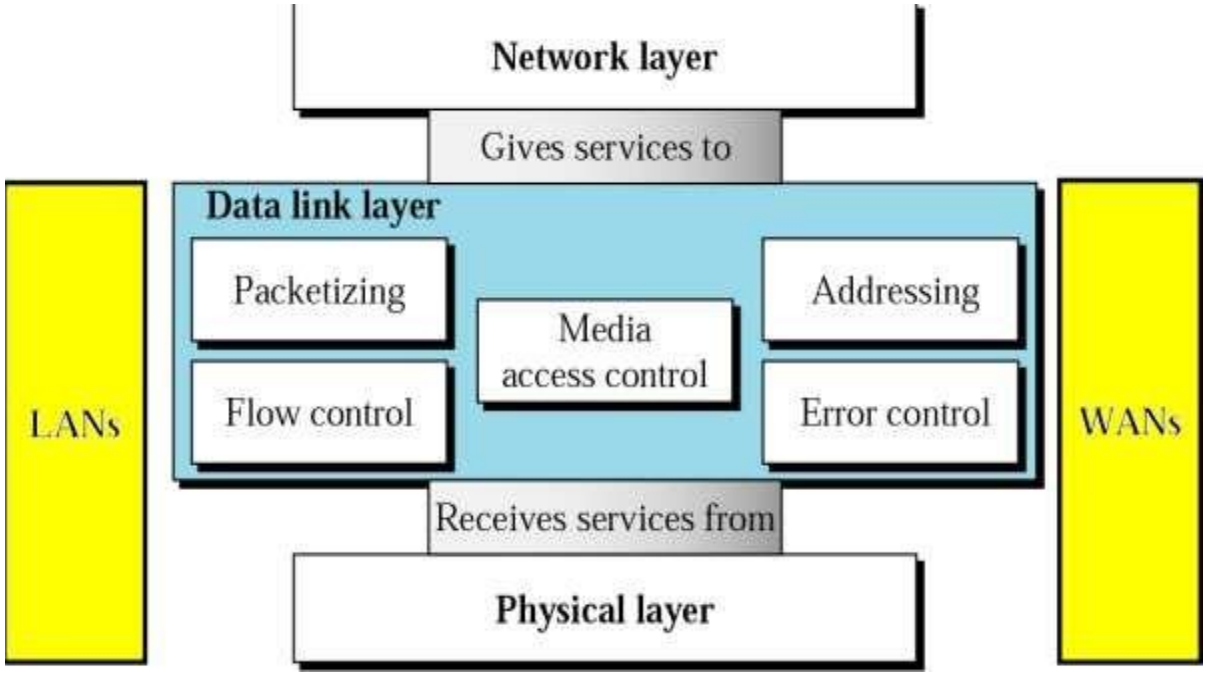
- Design issues
- Error detection& correction
- Elementary data link layer protocols
- Sliding window protocols

Multiple Access Protocols

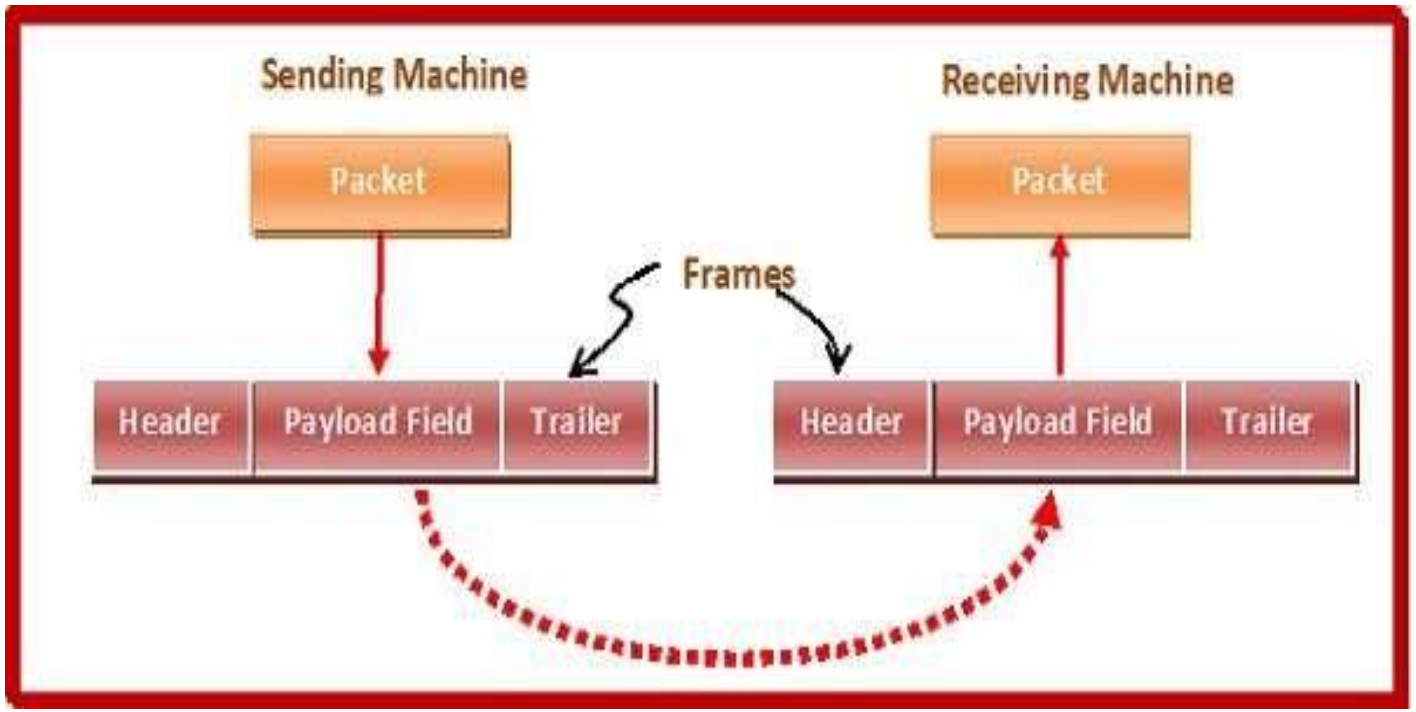
- ALOHA
- CSMA,CSMA/CD,CSMA/CA
- Collision free protocols
- Ethernet-Physical Layer
- Ethernet Mac Sub Layer
- Data link layer Switching
- Use of bridges
- Learning bridges
- Spanning tree bridges
- Repeaters
- Hubs
- Bridges
- Switches
- Routers
- Gate ways

OVERVIEW OF DLL

- The data link layer transforms the physical layer, a raw transmission facility, to a link responsible for node-to-node (hop-to-hop) communication. Specific responsibilities of the data link layer include addessing , framing , flow control , error control, media access control



For this, the data link layer takes the packets it gets from the network layer and encapsulates them into frames for transmission. Each frame contains a frame header, a payload field for holding the packet, and a frame trailer



Parts of a Frame

A frame has the following parts –

- **Frame Header** – It contains the source and the destination addresses of the frame.
- **Payload field** – It contains the message to be delivered.
- **Trailer** – It contains the error detection and error correction bits.
- **Flag** – It marks the beginning and end of the frame.

DLL DESIGN ISSUES

1. **Providing Services to the network layer:**

2. **Framing**

3. **Error Control**

4. **Flow Control**

1. SERVICES PROVIDED TO THE NETWORK LAYER

- The data link layer can be designed to offer various services.

The actual services offered can vary from system to system.

Three reasonable possibilities that are commonly provided are

1) Unacknowledged Connectionless service
2) Acknowledged Connectionless service
3) Acknowledged Connection-Oriented service

1.1 UNACKNOWLEDGED CONNECTIONLESS SERVICE

- Unacknowledged connectionless service consists of having the source machine send independent frames to the destination machine without having the destination machine acknowledge them.

- If a frame is lost due to noise on the line, no attempt is made to detect the loss or recover from it in the data link layer.

- This class of service is appropriate when the error rate is very low so that recovery is left to higher layers.

- Most LANs use unacknowledged connectionless service in the data link layer.

1.2 ACKNOWLEDGED CONNECTIONLESS SERVICE

- When this service is offered, there are still no logical connections used, but each frame sent is individually acknowledged.

- In this way, the sender knows whether a frame has arrived correctly. If it has not arrived within a specified time interval, it can be sent again. This service is useful over unreliable channels, such as wireless systems.

- If individual frames are acknowledged and retransmitted, entire packets get through much faster.

1.3 ACKNOWLEDGED CONNECTION-ORIENTED SERVICE

- Here, the source and destination machines establish a connection before any data are transferred. Each frame sent over the connection is numbered, and the data link layer guarantees that each frame sent is indeed received.

- Furthermore, it guarantees that each frame is received exactly once and that all frames are received in the right order.

Design issues: Services

Connectionless service

Unacknowledged

- Independent frames
- Error rate should be low
- Recovery left to higher layers
- Used on LANs

Acknowledged

- No connection
- Acknowledgement for each packet
- Resending
- Ack is optimisation; also transport layer can handle errors

Connection-oriented service

Acknowledged

- Each frame received exactly once
- All frames received in the right order
- 3 distinct phases:
  - Establishment of connection
  - Data transfer
  - Release of connection

2. FRAMING

- The usual approach is for the data link layer to break the bit stream up into discrete frames and compute the checksum for each frame (framing).

- When a frame arrives at the destination, **the checksum** is recomputed. If the newly computed checksum is different from the one contained in the frame, the data link layer knows that an error has occurred and takes steps to deal with it

- Example., discarding the bad frame and possibly also sending back an error report

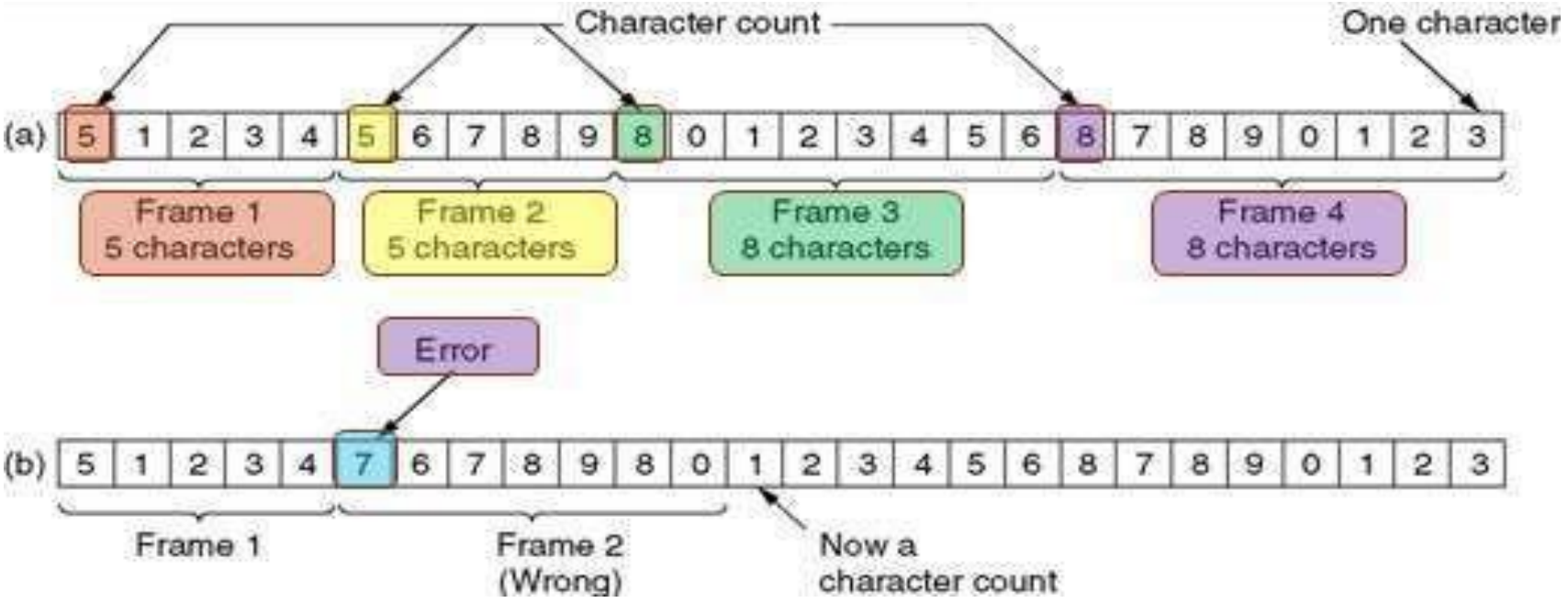
FRAMING(contd…)

We will look at four framing methods:

1. Character count.
2. Byte stuffing.
3. Bit stuffing.
4. Physical layer coding violations.

FRAMING – CHARACTER COUNT

- The first framing method **uses a field in the header** to specify the number of characters in the frame. When the data link layer at the destination sees the character count, it knows how many characters follow and hence where the end of the frame is. This technique is shown in fig a) four frames of sizes 5,5,8,8 characters respectively(without errors) fig.b) with errors



- The trouble with this algorithm is that the count can be garbled by a transmission error.
- **For example,** if the character count of 5 in the second frame of Fig. (b) becomes a 7, the destination will get out of synchronization and will be unable to locate the start of the next frame. Even if the checksum is incorrect so the destination knows that the frame is bad, it still has no way of telling where the next frame starts.
- Sending a frame back to the source asking for a retransmission does not help either, since the destination does not know how many characters to skip over to get to the start of the retransmission. For this reason, the character count method is rarely used anymore.

FRAMING – BYTE / CHARACTER STUFFING

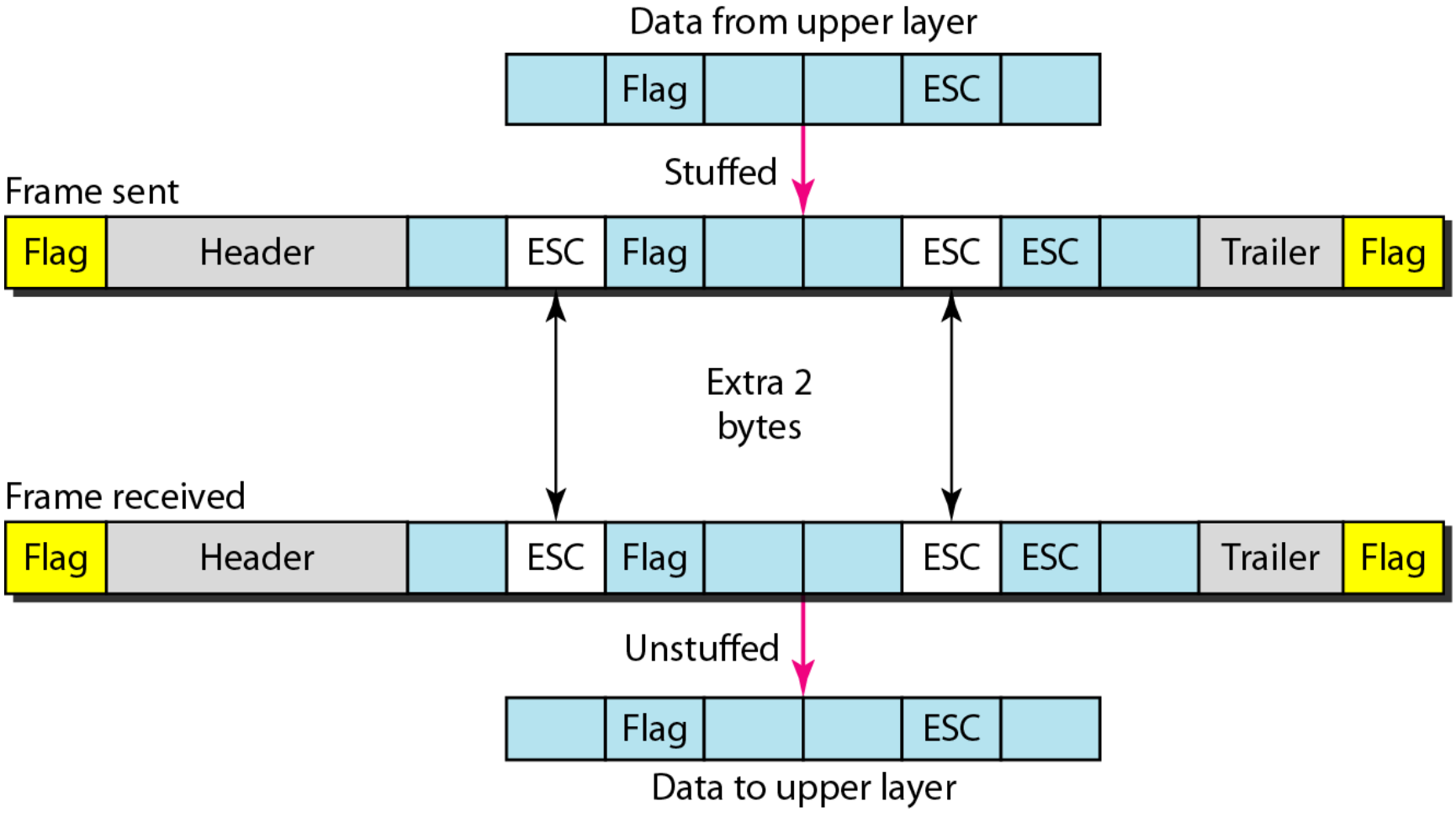
- In the past, the starting and ending bytes were different, but in recent years most protocols have used the same byte, called a flag byte, as both the starting and ending delimiter, as shown in below figure as FLAG.

- In this way, if the receiver ever loses synchronization, it can just search for the flag byte to find the end of the current frame. Two consecutive flag bytes indicate the end of one frame and start of the next one.

FRAMING – BYTE / CHARACTER STUFFING

- Each frame starts and ends with a FLAG byte. Thus adjacent frames are separated by two flag bytes.
- A serious problem occurs with this method is when binary data is transmitted ,It is possible that FLAG is actually a part of the data.

- **Solution:** At the sender an escape byte (ESC) character is inserted just before the FLAG byte present in the data. The data link layer at the receiver end removes the ESC is from the data before sending it to the network layer. This technique is called as byte stuffing or character stuffing.
- Thus , a framing flag bye can be distinguished from one in the data by absence or presence of an escape byte before it.
- Now if an ESC is present in the data then an extra ESC is inserted before it in the data. This extra ESC is removed at the receiver.

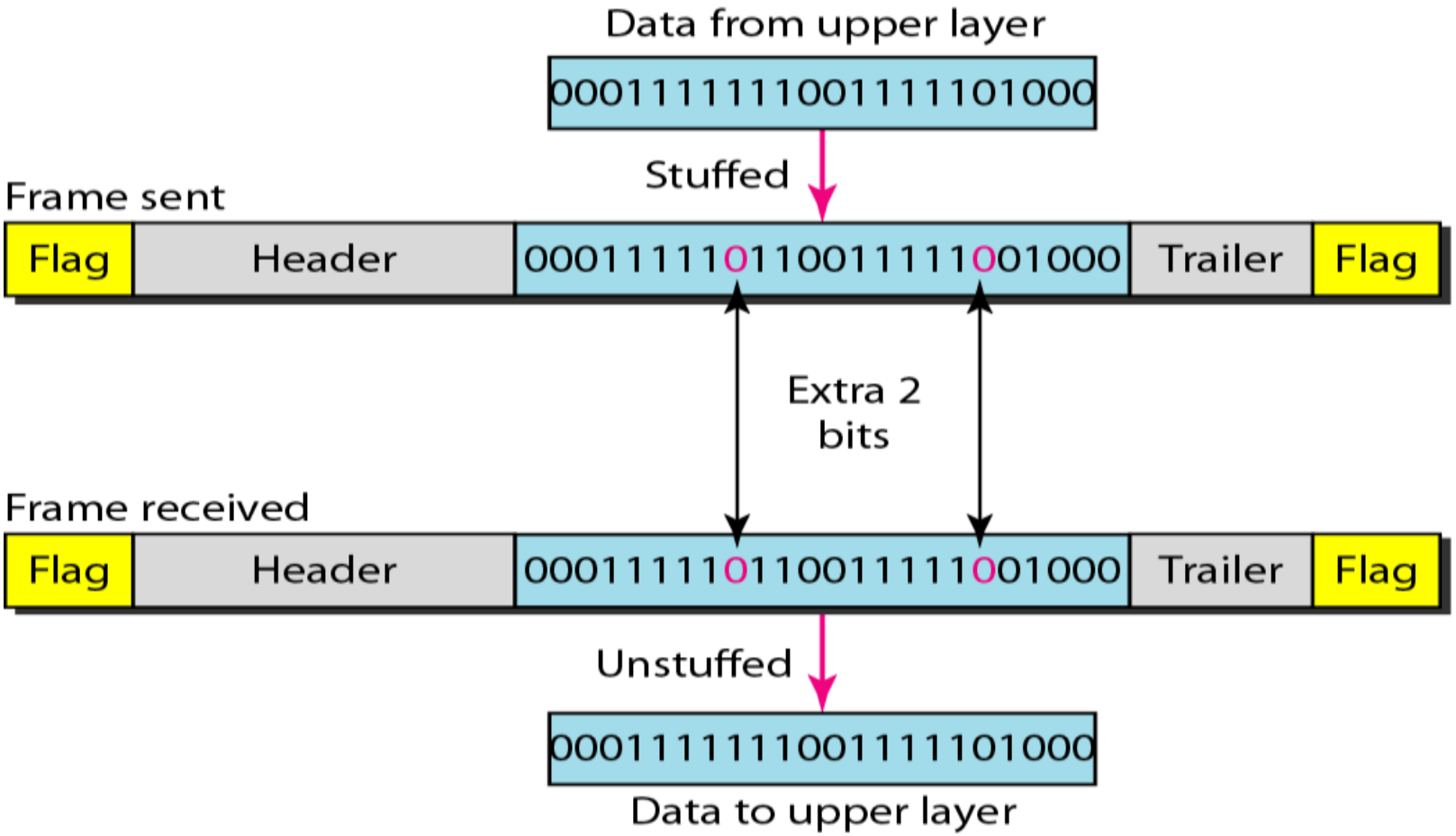


The major disadvantage of using this framing method is that it is closely tied to the use of 8-bit characters.

FRAMING – BIT STUFFING

- Whenever the sender's data link layer encounters five consecutive 1s in the data, it automatically stuffs a 0 bit into the outgoing bit stream.
- This bit stuffing is analogous to byte stuffing , in which an escape bye is stuffed into the outgoing character stream before a flag byte in the data.
- When the receiver sees five consecutive incoming 1 bits, followed by a 0 bit, it automatically de- stuffs (i.e., deletes) the 0 bit. Just as byte stuffing is completely transparent to the network layer in both computers, so is bit stuffing.

FRAMING – BIT STUFFING



Types of Errors

Single bit error-

Multiple bits error- more than one bits in corrupted state**.**

Burst error-Frame contains more than1 consecutive bits corrupted.

3. ERROR CONTROL

Error control mechanism may involve two possible ways:

- **Error Detection**
- **Error Correction**

Error Detection

- Error detection means to decide whether the received data is correct or not without having a copy of the original message.

REDUNDANCY

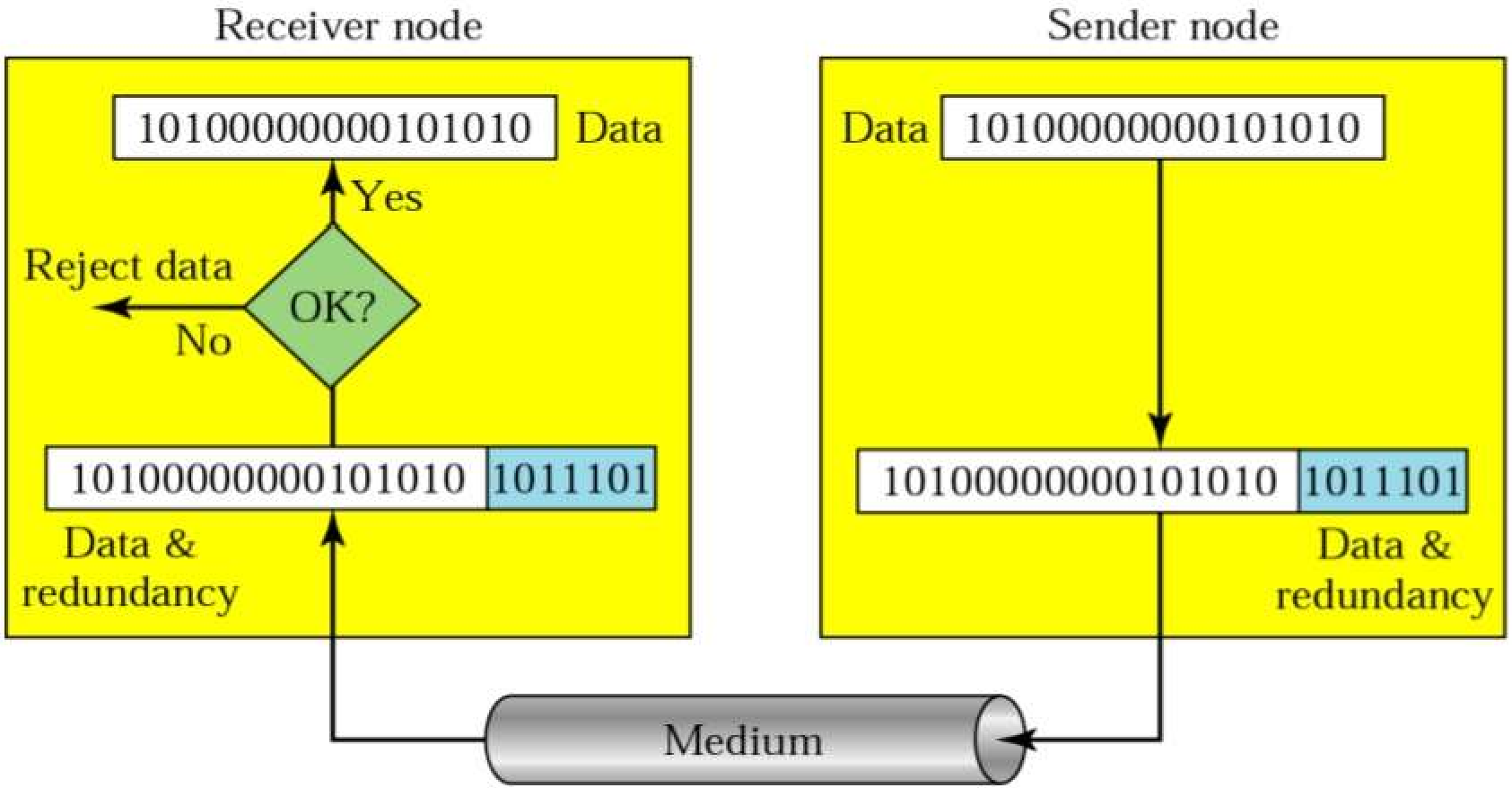
- The concept of including extra information in the transmission for error detection is called redundancy.
- Instead of repeating the entire data stream,a shorter group of bits may be appended to the end of each unit.
- The technique is called redundancy because the extra bits are redundant to the information.

REDUNDANCY

- These extra bits are discarded as soon as the accuracy of the transmission has been determined.

- All Error Detection mechanisms use the concept of redundancy which means adding extra bits for detecting errors.

REDUNDANCY

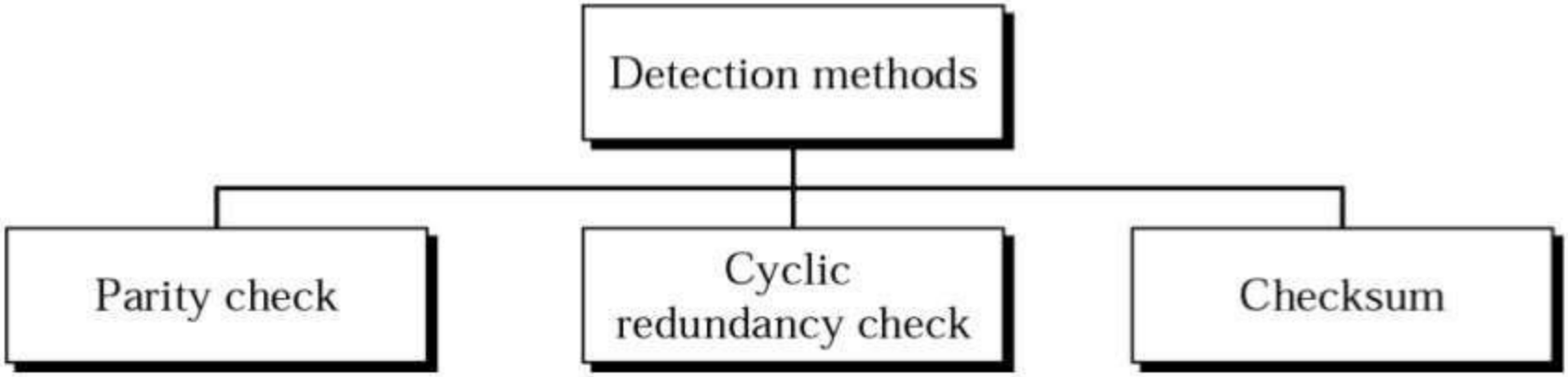

Receiver node
Sender node
1010000000101010
Data
Yes
Reject data
No
OK?
1010000000101010
1011101
Data &
redundancy
Data
1010000000101010
1010000000101010
1011101
Data &
redundancy
Medium

Error Detecting Techniques:

The most popular Error Detecting Techniques are:

- Single parity check
- Two-dimensional parity check
- Checksum
- Cyclic redundancy check

ERROR DETECTION METHODS



PARITY CHECK

- The most common and least expensive mechanism for error detection is the parity check.

- Parity checking can be
  - simple

    OR
  - two-dimensional.

SIMPLE PARITY CHECK
OR
VERTICAL REDUNDANCY CHECK(VRC)

- In this technique, the redundant bit called a parity bit.
- A parity bit is added to every data unit so that the total number of 1s in the unit(including the parity bit) becomes even(or odd).

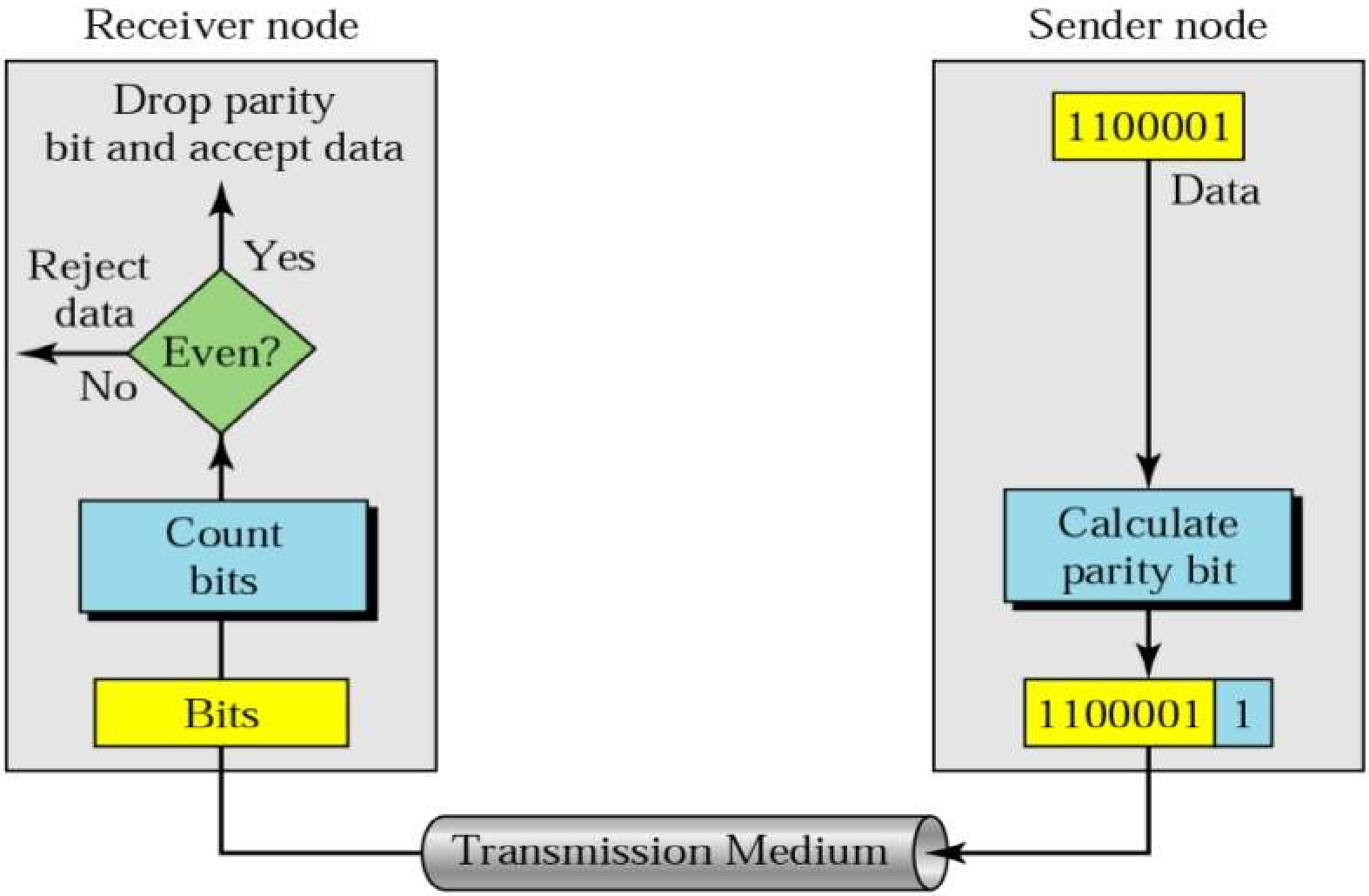
Single Parity Check

- Single Parity checking is the simple mechanism and inexpensive to detect the errors.
- In this technique, a redundant bit is also known as a parity bit which is appended at the end of the data unit so that the number of 1s becomes even.
- If the number of 1s bits is odd, then parity bit 1 is appended and if the number of 1s bits is even, then parity bit 0 is appended at the end of the data unit.
- At the receiving end, the parity bit is calculated from the received data bits and compared with the received parity bit.
- This technique generates the total number of 1s even, so it is known as even-parity checking.

PARITY CHECK

- One extra bit is sent along with the original bits to make number of 1s either even in case of even parity, or odd in case of odd parity.
- The sender while creating a frame counts the number of 1s in it. For example, if even parity is used and number of 1s is even then one bit with value 0 is added.
- This way number of 1s remains even.If the number of 1s is odd, to make it even a bit with value 1 is added.

EVEN PARITY CONCEPT



Example 1

Suppose the sender wants to send the word *world*. In ASCII the five characters are coded as

1110111 1101111 1110010 1101100 1100100

w o r l d

The following shows the actual bits sent

11101110 11011110 11100100 11011000 11001001

EXAMPLE 1 CONTINUED...

- Now suppose the word world in Example 1 is received by the receiver without being corrupted in transmission.

- 11101110 11011110 11100100 11011000 11001001
- w o r l d
- The receiver counts the 1s in each character and comes up with even numbers (6, 6, 4, 4, 4). The data are accepted.

EXAMPLE 1 CONTINUED…

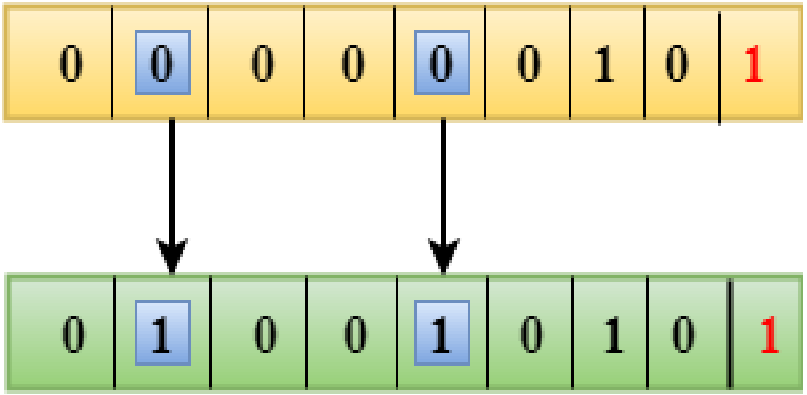
- Now suppose the word world in Example 1 is corrupted during transmission.
- 111**1**1110 11011110 1110**1**100 11011000 11001001
- The receiver counts the 1s in each character and comes up with even and odd numbers (7, 6, 5, 4, 4). The receiver knows that the data are corrupted, discards them, and asks for retransmission.

PERFORMANCE OF VRC OR SIMPLE PARITY CHECK

- Simple parity check can detect all single-bit errors.
- It can also detect burst errors as long as the total number of bits changed is odd.
- It cannot detect burst errors where the number of bits changed is even.

Drawbacks Of Single Parity Checking

- It can only detect single-bit errors which are very rare.
- If two bits are interchanged, then it cannot detect the errors.



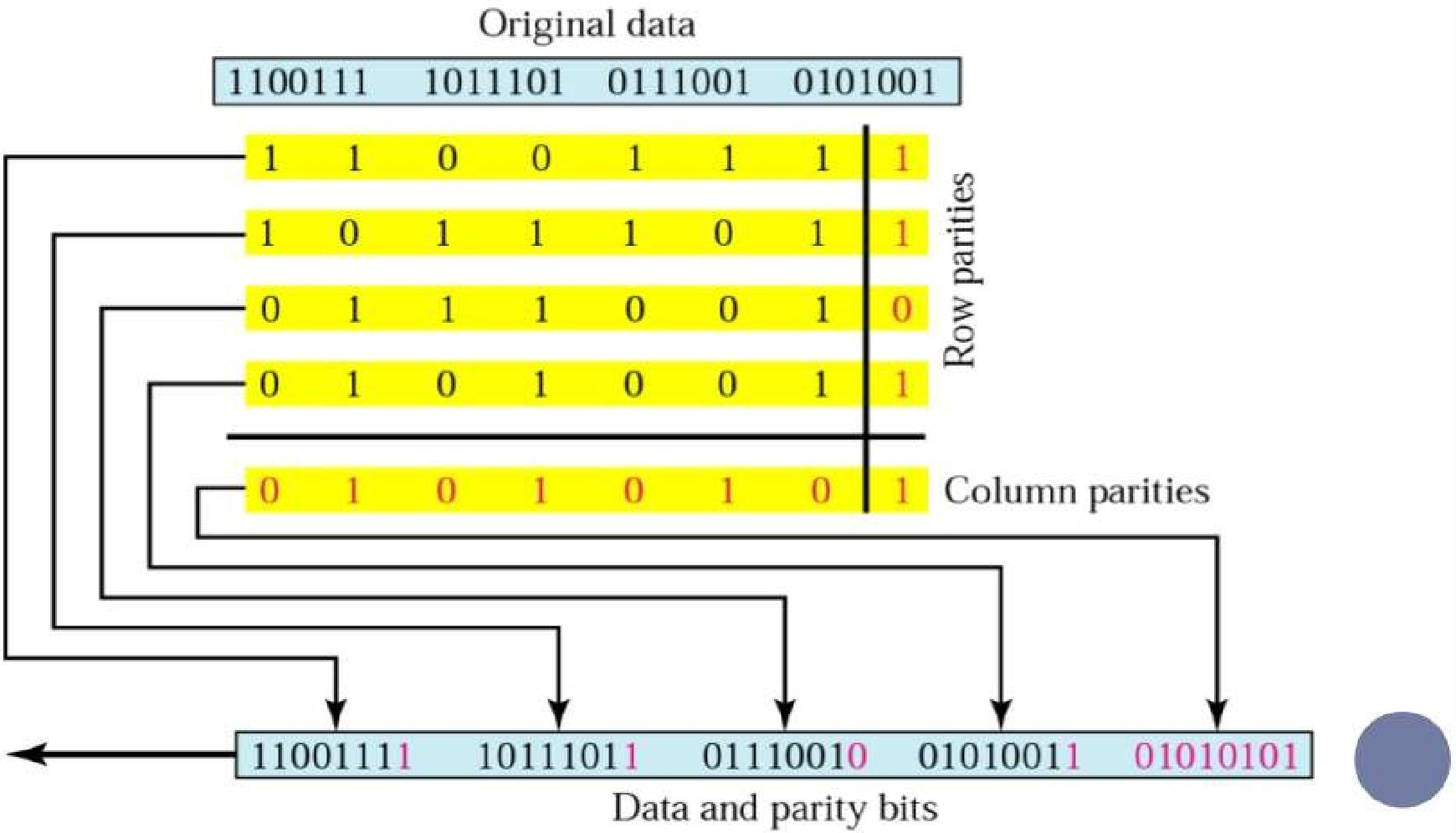
Two-Dimensional Parity Check:

Performance can be improved by using **Two-Dimensional Parity Check** which organizes the data in the form of a table.

TWO DIMENSIONAL PARITY CHECK OR LONGITUDINAL REDUNDANCY CHECK

- In this method a block of bits is organized in a table(rows and columns).
- First we calculate the parity bit for each data unit.
- Then the data is organized in to a table.
- Then a parity bit for each column is calculated.
- This creates a new row.
- This new row contains the parity bits for the whole block.
- Then this new row is attached to the original data and sent to the receiver.

TWO-DIMENSIONAL PARITY OR LRC



Drawbacks Of 2D Parity Check:

- If two bits in one data unit are corrupted and two bits exactly the same position in another data unit are also corrupted, then 2D Parity checker will not be able to detect the error.
- This technique cannot be used to detect the 4-bit errors or more in some cases.

Cyclic Redundancy Check (CRC)

- CRC is a different approach to detect if the received frame contains valid data. This technique involves binary division of the data bits being sent.

- The divisor is generated using polynomials. The sender performs a division operation on the bits being sent and calculates the remainder.

- Before sending the actual bits, the sender adds the remainder at the end of the actual bits. Actual data bits plus the remainder is called a code word. The sender transmits data bits as codewords.

CRC is a redundancy error technique used to determine the error.

Following are the steps used in CRC for error detection:

•In CRC technique, a string of n 0s is appended to the data unit, and this n number is less than the number of bits in a predetermined number, known as division which is n+1 bits.

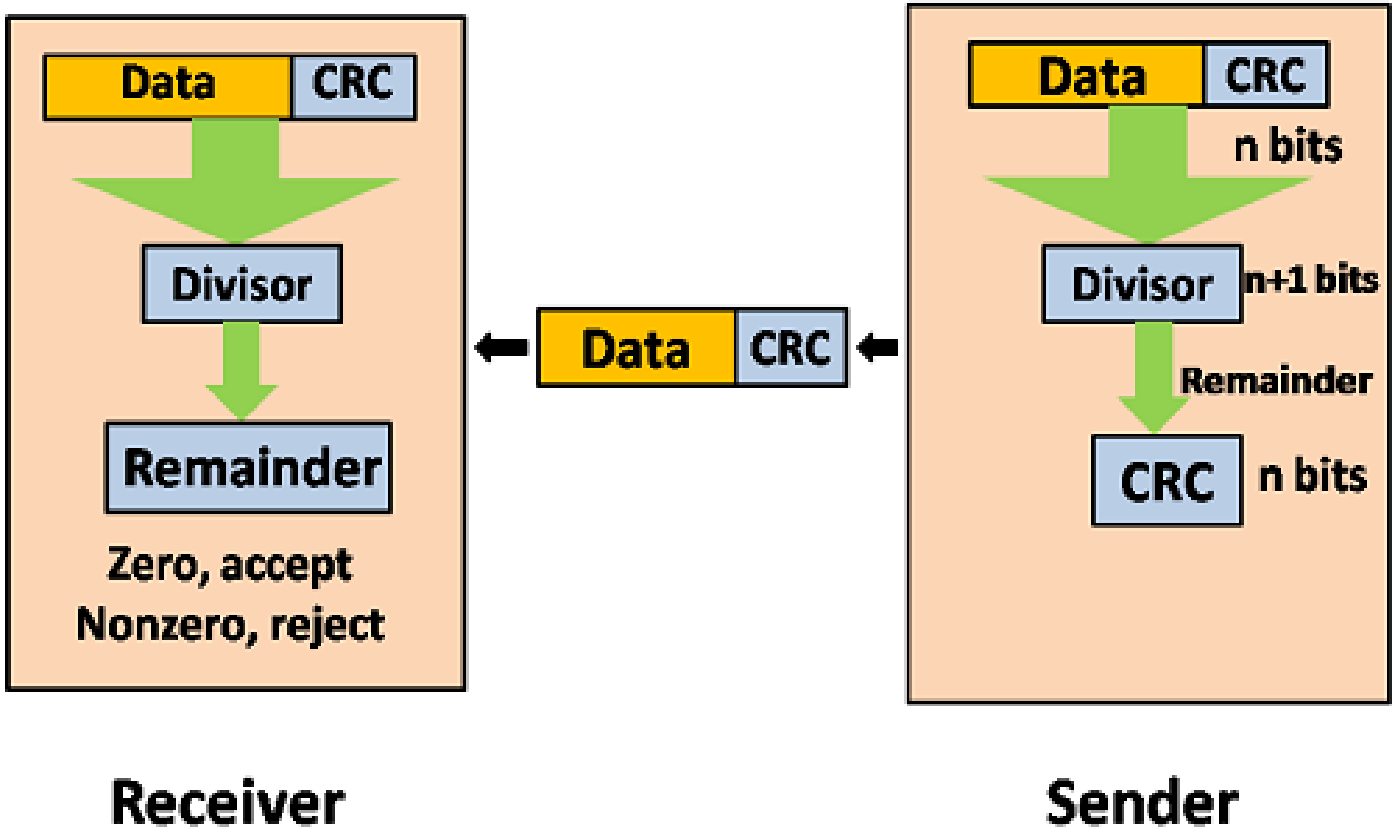
•Secondly, the newly extended data is divided by a divisor using a process is known as binary division. The remainder generated from this division is known as CRC remainder.

•Thirdly, the CRC remainder replaces the appended 0s at the end of the original data. This newly generated unit is sent to the receiver.

•The receiver receives the data followed by the CRC remainder. The receiver will treat this whole unit as a single unit, and it is divided by the same divisor that was used to find the CRC remainder.

•If the resultant of this division is zero which means that it has no error, and the data is accepted.

•If the resultant of this division is not zero which means that the data consists of an error. Therefore, the data is discarded.


Data
CRC
Divisor
Remainder
Zero, accept
Nonzero, reject
Data
CRC
Data
CRC
n bits
Divisor
n+1 bits
Remainder
CRC
n bits
Receiver
Sender

Example1

- A bit stream **1001** is transmitted using the standard CRC method. The generator polynomial is **x^3+x+1**.

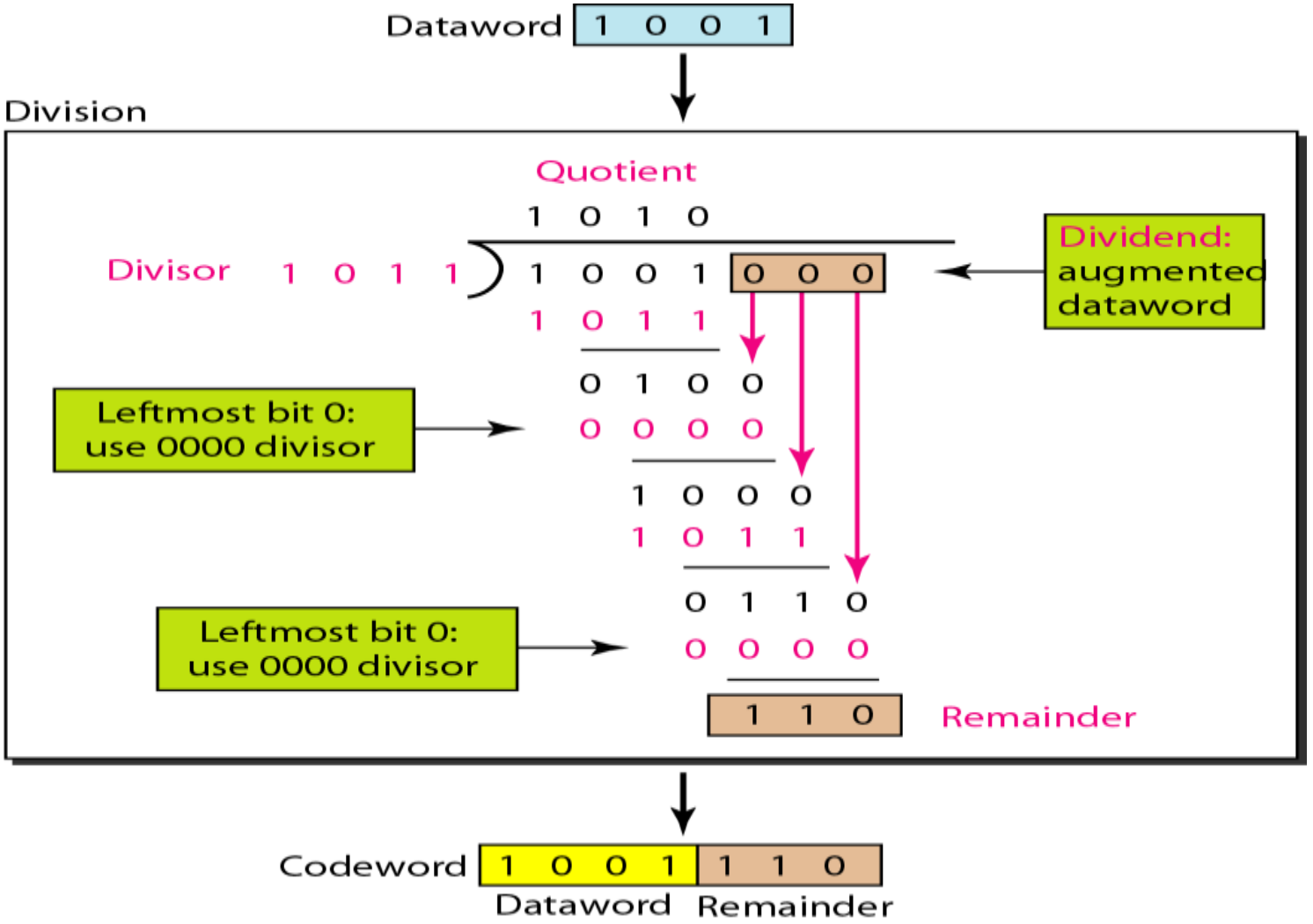
  • The generator polynomial G(x) = x3 + x+1 is encoded as **1011 - divisor**

  • Clearly, the generator polynomial consists of 4 bits.

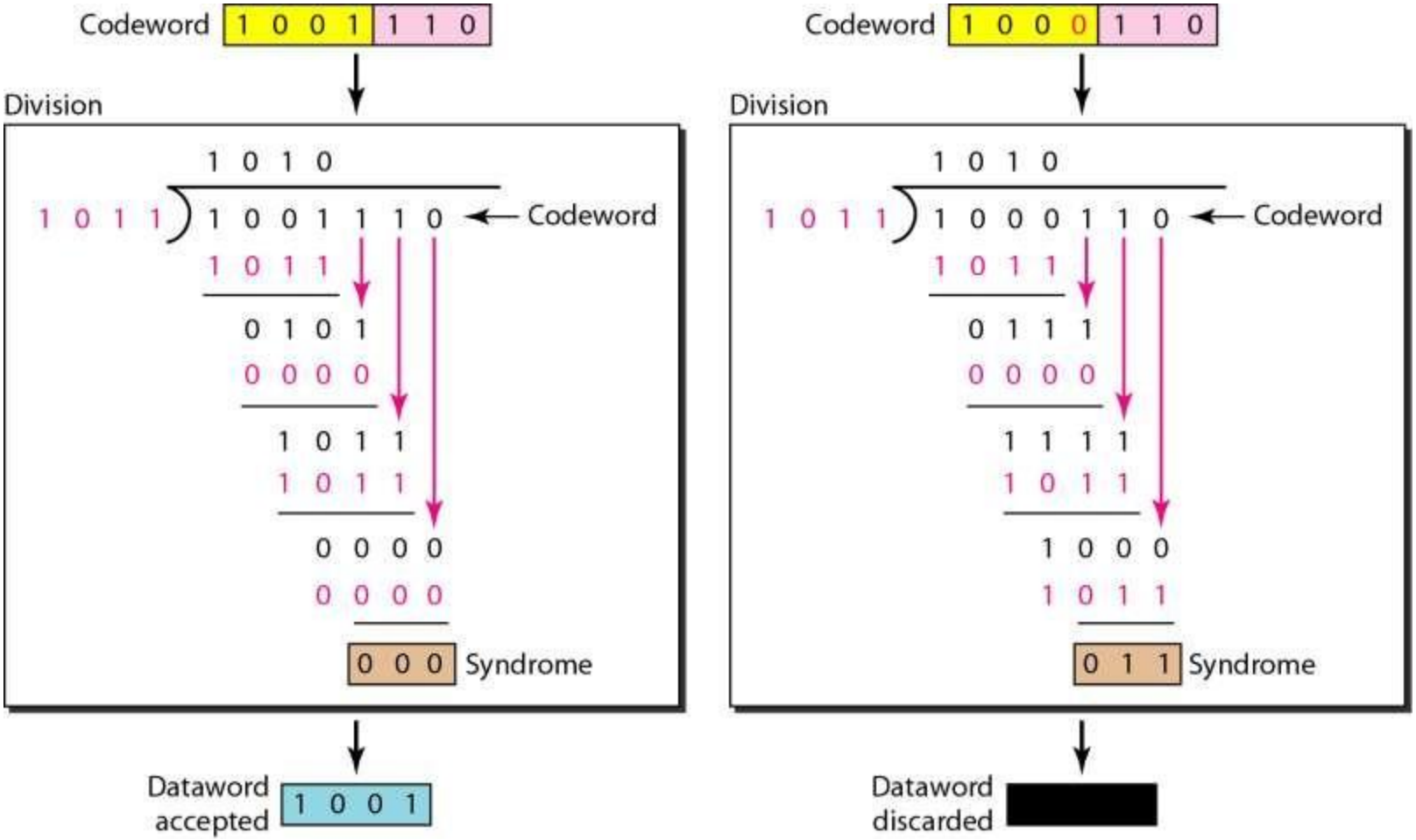
  • So, a string of 3 zeroes is appended to the bit stream to be transmitted.

  • The resulting bit stream is **1001000 - dividend**

Division in CRC encoder



Division in the CRC decoder for two cases

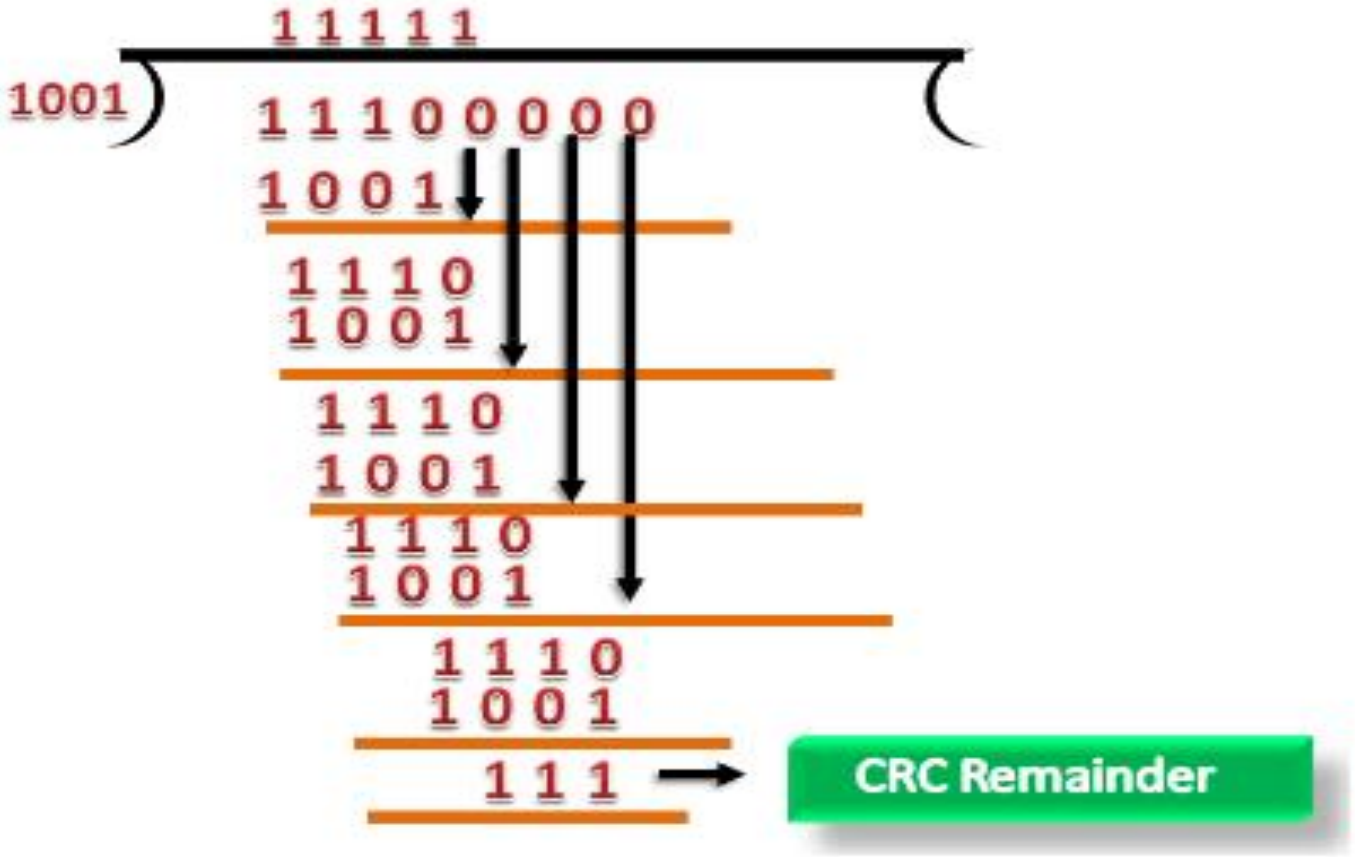


Example 2:

Suppose the original data is 11100 and divisor is 1001.

CRC Generator

- A CRC generator uses a modulo-2 division. Firstly, three zeroes are appended at the end of the data as the length of the divisor is 4 and we know that the length of the string 0s to be appended is always one less than the length of the divisor.
- Now, the string becomes 11100000, and the resultant string is divided by the divisor 1001.
- The remainder generated from the binary division is known as CRC remainder. The generated value of the CRC remainder is 111.
- CRC remainder replaces the appended string of 0s at the end of the data unit, and the final string would be 11100111 which is sent across the network.


11111
1001
11100000
1001
1110
1001
1110
1001
1110
1001
1110
1001
111
CRC Remainder

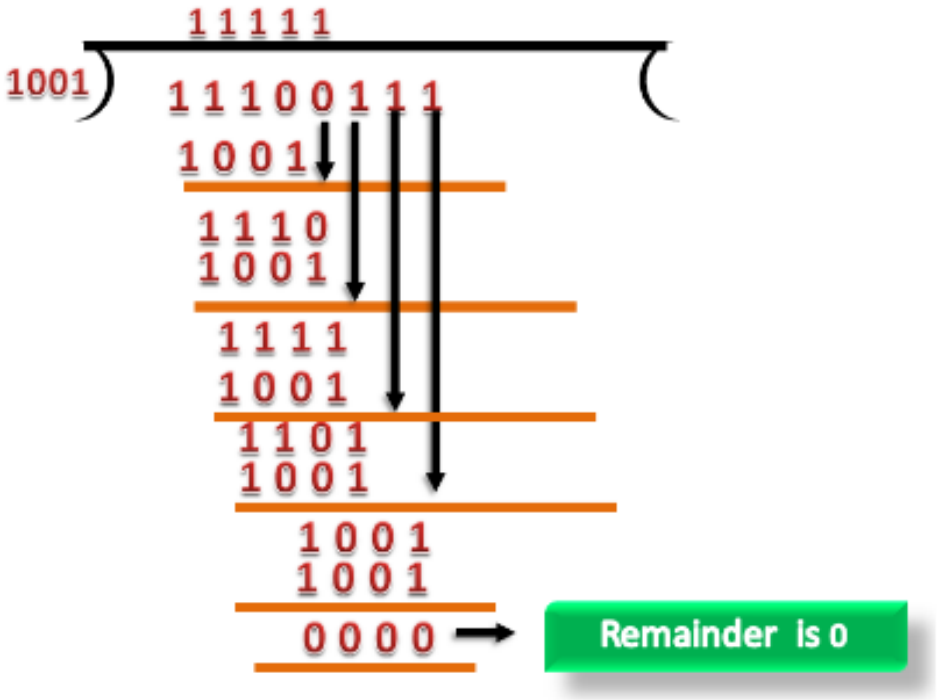
CRC Checker

•The functionality of the CRC checker is similar to the CRC generator.

•When the string 11100111 is received at the receiving end, then CRC checker performs the modulo-2 division.

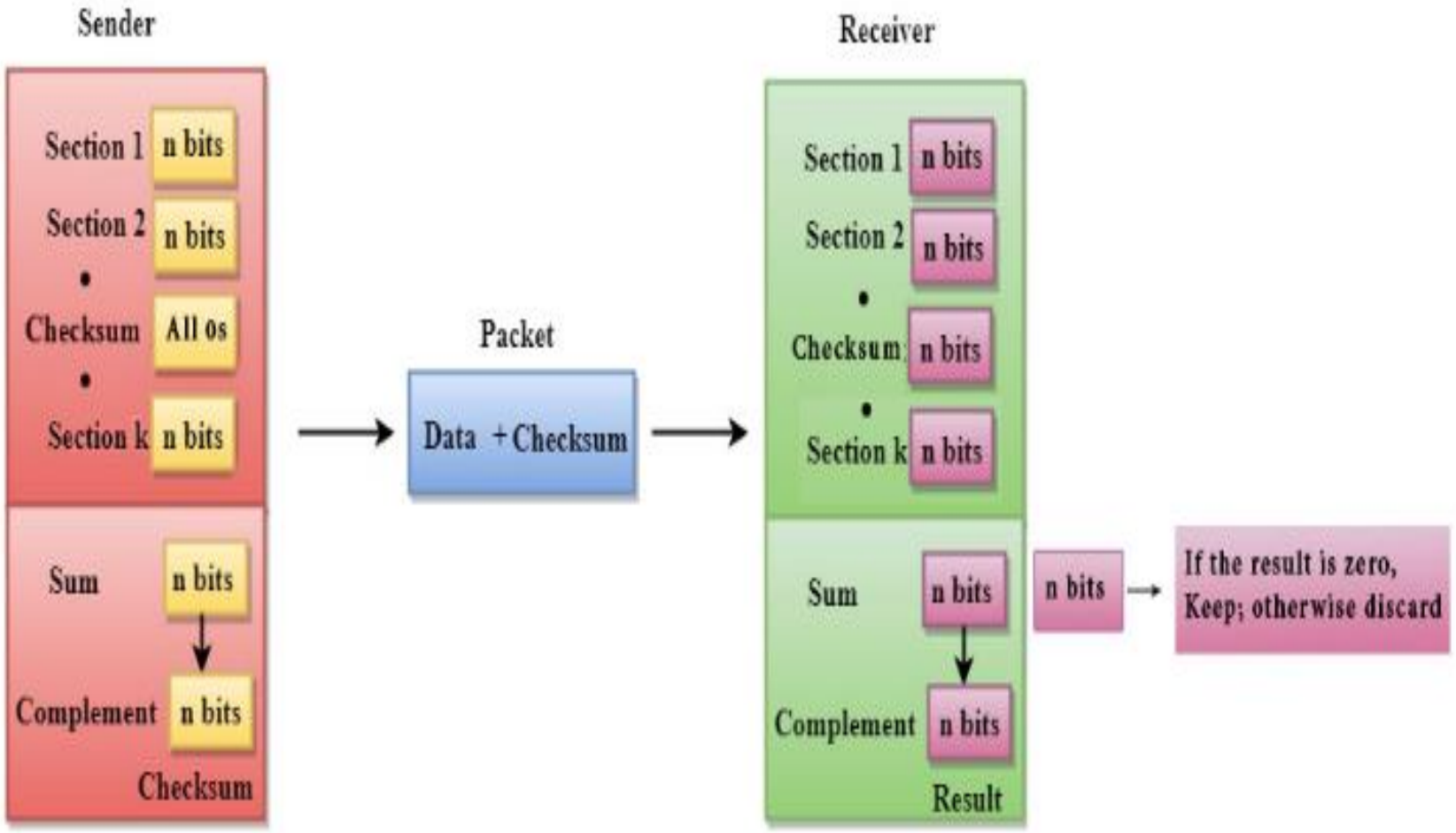
•A string is divided by the same divisor, i.e., 1001.

•In this case, CRC checker generates the remainder of zero. Therefore, the data is accepted.



CHECKSUM

- In checksum error detection scheme, the data is divided into k segments each of m bits.
- In the sender's end the segments are added using 1's complement arithmetic to get the sum. The sum is complemented to get the checksum.
- The checksum segment is sent along with the data segments.
- At the receiver's end, all received segments are added using 1's complement arithmetic to get the sum. The sum is complemented.


Sender
Section 1
n bits
Section 2
n bits
Checksum
All 0s
Section k
n bits
Sum
n bits
Complement
n bits
Checksum
Packet
Data + Checksum
Receiver
Section 1
n bits
Section 2
n bits
Checksum
n bits
Section k
n bits
Sum
n bits
Complement
n bits
Result
n bits
If the result is zero,
Keep; otherwise discard

At Sender

The Sender follows the given steps:

- The block unit is divided into k sections, and each of n bits.
- All the k sections are added together by using one's complement to get the sum.
- The sum is complemented and it becomes the checksum field.
- The original data and checksum field are sent across the network.

At Receiver

The Receiver follows the given steps:

- The block unit is divided into k sections and each of n bits.
- All the k sections are added together by using one's complement algorithm to get the sum.
- The sum is complemented.
- If the result of the sum is zero, then the data is accepted otherwise the data is discarded.

Example

- Suppose that the sender wants to send 4 frames each of 8 bits, where the frames are 11001100, 10101010, 11110000 and 11000011.
- The sender adds the bits using 1s complement arithmetic. While adding two numbers using 1s complement arithmetic, if there is a carry over, it is added to the sum.
- After adding all the 4 frames, the sender complements the sum to get the checksum, 11010011, and sends it along with the data frames.
- The receiver performs 1s complement arithmetic sum of all the frames including the checksum. The result is complemented and found to be 0. Hence, the receiver assumes that no error has occurred.

Sender's End

Frame 1:	11001100
Frame 2:	+ 10101010
Partial Sum:	1 01110110
	+ 1
	01110111
Frame 3:	+ 11110000
Partial Sum:	1 01100111
	+ 1
	01101000
Frame 4:	+ 11000011
Partial Sum:	1 00101011
	+ 1
Sum:	00101100
Checksum:	11010011

Receiver's End

Frame 1:	11001100
Frame 2:	+ 10101010
Partial Sum:	1 01110110
	+ 1
	01110111
Frame 3:	+ 11110000
Partial Sum:	1 01100111
	+ 1
	01101000
Frame 4:	+ 11000011
Partial Sum:	1 00101011
	+ 1
Sum:	00101100
Checksum:	11010011
Sum:	11111111
Complement:	00000000

Hence accept frames.

HAMMING CODE ERROR DETECTION & CORRECTION

- The hamming code technique, which is an error-detection and error-correction technique, was proposed by R.W. Hamming.

- Whenever a data packet is transmitted over a network, there are possibilities that the data bits may get lost or damaged during transmission.

- Hamming codes make use of multiple parity bits to enable single bit correction and two-bit error detection

Hamming Code = Data Bits + Parity Bits

Hamming Code

Hamming code is used to detect and correct the error in the transmitted data. So, it is an error detection and correction code. It was originally invented by *Richard W. Hamming* in the year 1950. Hamming codes detect 1-bit and 2-bit errors.

Parity bits: The bit which is appended to the original data of binary bits so that the total number of 1s is even or odd.

Even parity: To check for even parity, if the total number of 1s is even, then the value of the parity bit is 0. If the total number of 1s occurrences is odd, then the value of the parity bit is 1.

Odd Parity: To check for odd parity, if the total number of 1s is even, then the value of parity bit is 1. If the total number of 1s is odd, then the value of parity bit is 0.

•While transmitting the message, it is encoded with the redundant bits. The redundant bits are the extra bits that are placed at certain locations of the data bits to detect the error. At the receiver end, the code is decoded to detect errors and the original message is received.

•So before transmitting, the sender has to encode the message with the redundant bits. It involves three steps, as described below.

Encoding the message with hamming code

1. Selecting the number of redundant bits

•The hamming code uses the number of redundant bits depending on the number of information bits in the message.

•Let n be the number of information or data bits, then the number of redundant bits P is determined from the following formula,

$$2^P \geq n + P + 1$$

So the number of redundant bits is determined by the trial and error method.

2. Choosing the location of redundant bits

For example, the number of data bits n=4, and the number of redundant bits P=3. So the message consists of 7 bits in total that are to be coded. Let the rightmost bit be designated as bit 1, the next successive bit as bit 2 and so on. The seven bits are bit 7, bit 6, bit 5, bit 4, bit 3, bit 2, bit 1.

In this, the redundant bits are placed at the positions that are numbered corresponding to the power of 2, i.e., 1, 2, 4, 8,… Thus the locations of data bit and redundant bit are *D4, D3, D2, P3, D1, P2, P1*.

3. Assigning the values to redundant bits

•Now it is time to assign bit value to the redundant bits in the formed hamming code group. The assigned bits are called a parity bit.

• parity bit will check certain other bits in the total code group. It is one with the bit location table, as shown below.

Bit Location	7	6	5	4	3	2	1
Bit designation	D4	D3	D2	**P3**	D1	**P2**	**P1**
Binary representation	111	110	101	100	011	010	001
Information / Data bits	D4	D3	D2		D1		
Parity bits				**P3**		**P2**	**P1**

•Parity bit P1 covers all data bits in positions whose binary representation has 1 in the least significant position(001, 011, 101, 111, etc.). Thus P1 checks the bit in locations 1, 3, 5, 7, 9, 11, etc..

•Parity bit P2 covers all data bits in positions whose binary representation has 1 in the second least significant position(010, 011, 110, 111, etc.). Thus P2 checks the bit in locations 2, 3, 6, 7, etc.

•Parity bit P3 covers all data bits in positions whose binary representation has 1 in the third least significant position(100, 101, 110, 111, etc.). Thus P3 checks the bit in locations 4, 5, 6, 7, etc.

•Each parity bit checks the corresponding bit locations and assign the bit value as 1 or 0, so as to make the number of 1s as even for even parity and odd for odd parity.

Example problem 1

Encode a binary word 11001 into the even parity hamming code.

Given, number of data bits, n =5.

To find the number of redundant bits,

Let us try P=4.

$$2^4 \geq 5 + 4 + 1$$

The equation is satisfied and so 4 redundant bits are selected.

So, total code bit = n+P = 9

The redundant bits are placed at bit positions 1, 2, 4 and 8.

Construct the bit location table.

Bit Location	9	8	7	6	5	4	3	2	1
Bit designation	D5	**P4**	D4	D3	D2	**P3**	D1	**P2**	**P1**
Binary representation	1001	1000	0111	0110	0101	0100	0011	0010	0001
Information bits	1		1	0	0		1		
Parity bits		1				1		0	1

To determine the parity bits

- For P1: Bit locations 3, 5, 7 and 9 have three 1s. To have even parity, P1 must be 1.

- For P2: Bit locations 3, 6, 7 have two 1s. To have even parity, P2 must be 0.

- For P3: Bit locations 5, 6, 7 have one 1s. To have even parity, P3 must be 1.

- For P4: Bit locations 8, 9 have one 1s. To have even parity, P2 must be 1.

- Thus the encoded 9-bit hamming code is 111001101.

How to detect and correct the error in the hamming code?

•After receiving the encoded message, each parity bit along with its corresponding group of bits are checked for proper parity. While checking, the correct result of individual parity is marked as 0 and the wrong result is marked as 1.

•After checking all the parity bits, a binary word is formed taking the result bits for P1 as LSB. So formed binary word gives the bit location, where there is an error.

•If the formed binary word has 0 bits, then there is no error in the message.

Example problem 2

Let us assume the even parity hamming code from the above example (111001101) is transmitted and the received code is (110001101). Now from the received code, let us detect and correct the error.

To detect the error, let us construct the bit location table.

Bit Location	9	8	7	6	5	4	3	2	1
Bit designation	D_5	**P_4**	D_4	D_3	D_2	**P_3**	D_1	**P_2**	**P_1**
Binary representation	1001	1000	0111	0110	0101	0100	0011	0010	0001
Received code	1	1	0	0	0	1	1	0	1

Checking the parity bits

•For P1 : Check the locations 1, 3, 5, 7, 9. There is three 1s in this group, which is wrong for even parity. Hence the bit value for P1 is 1.

•For P2 : Check the locations 2, 3, 6, 7. There is one 1 in this group, which is wrong for even parity. Hence the bit value for P2 is 1.

•For P3 : Check the locations 3, 5, 6, 7. There is one 1 in this group, which is wrong for even parity. Hence the bit value for P3 is 1.

•For P4 : Check the locations 8, 9. There are two 1s in this group, which is correct for even parity. Hence the bit value for P4 is 0.

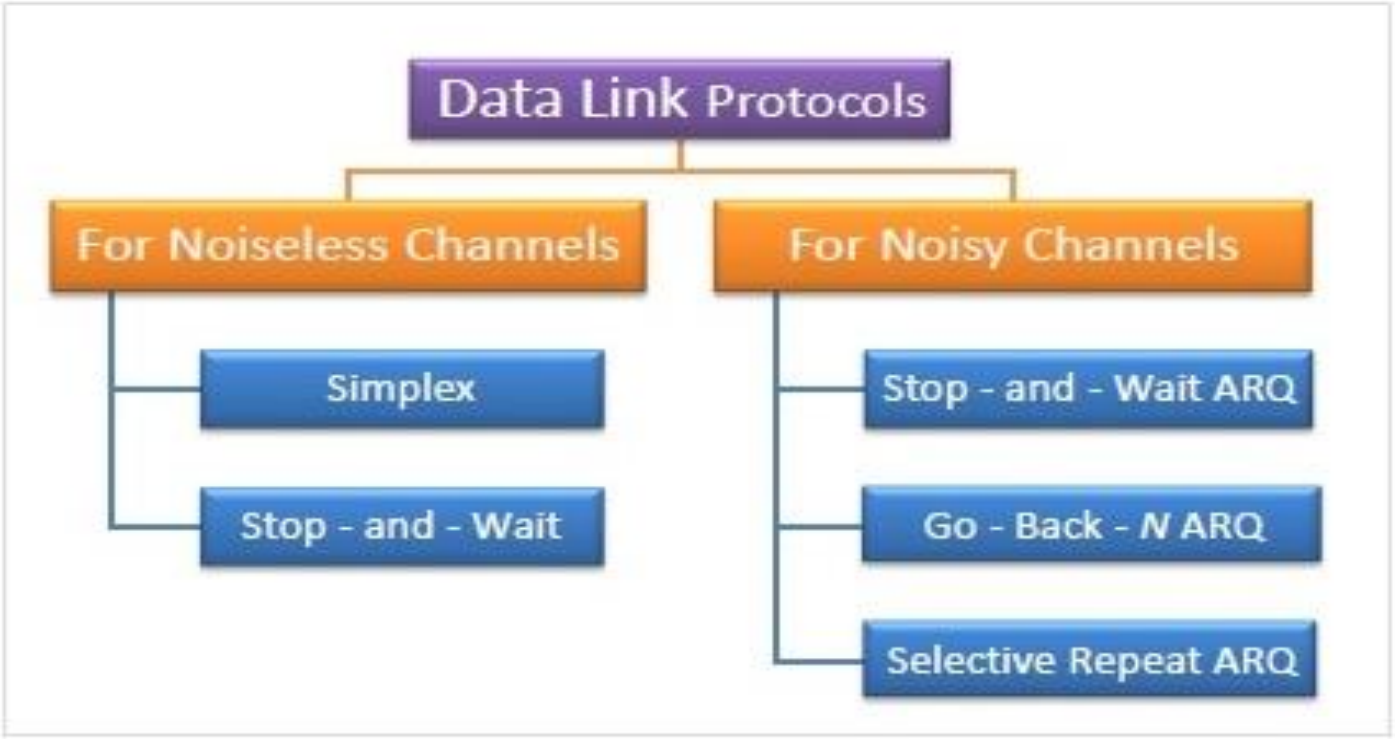
•The resultant binary word is 0111. It corresponds to the bit location 7 in the above table. The error is detected in the data bit D4. The error is 0 and it should be changed to 1. **Thus the corrected code is 111001101**.

ELEMENTARY DATA LINK LAYER PROTOCOLS

- Protocols in the data link layer are designed so that this layer can perform its basic functions: framing, error control and flow control.
- Framing is the process of dividing bit - streams from physical layer into data frames whose size ranges from a few hundred to a few thousand bytes.
- Error control mechanisms deals with transmission errors and retransmission of corrupted and lost frames.
- Flow control regulates speed of delivery and so that a fast sender does not drown a slow receiver.

Types of Data Link Protocols

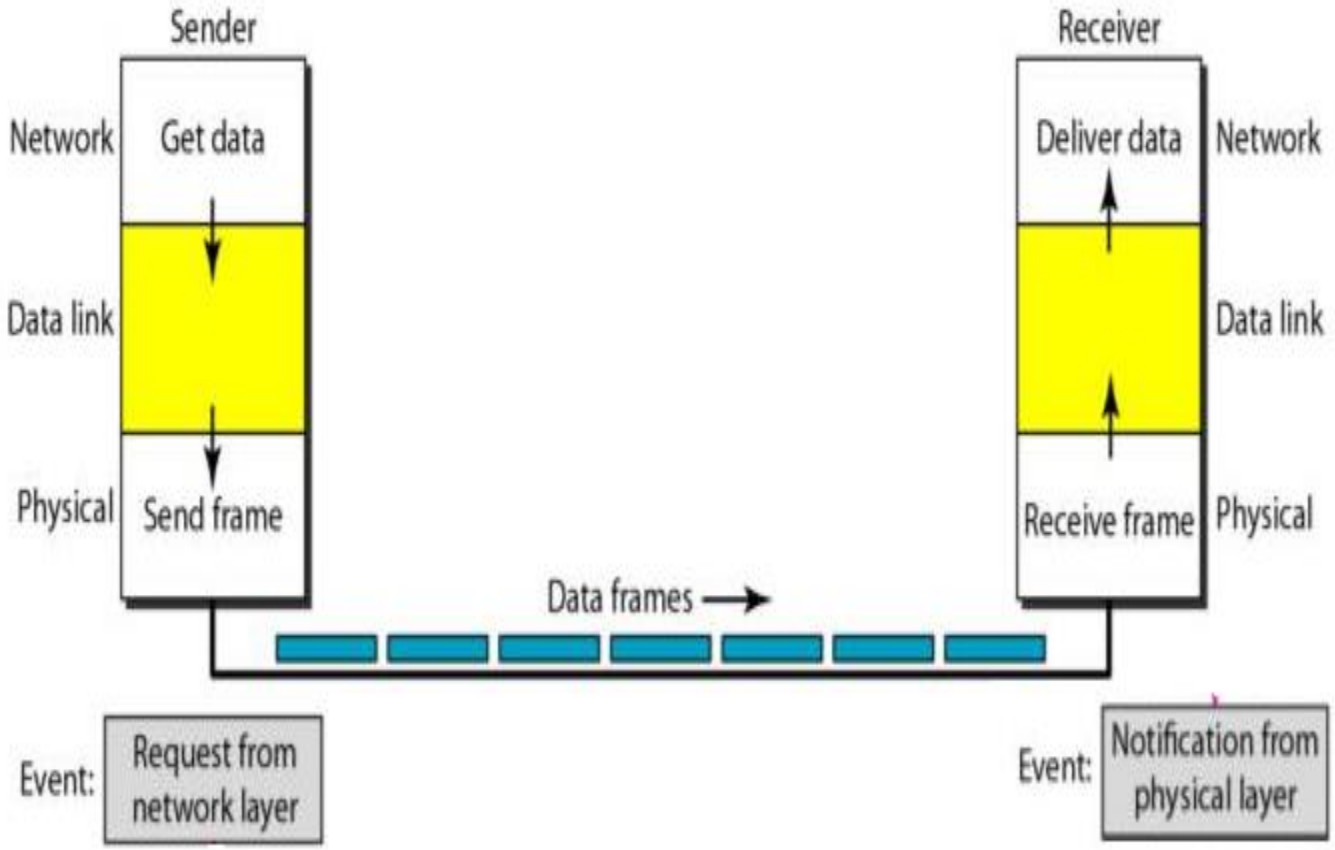
Data link protocols can be broadly divided into two categories, depending on whether the transmission channel is noiseless or noisy.



Simplest Protocol

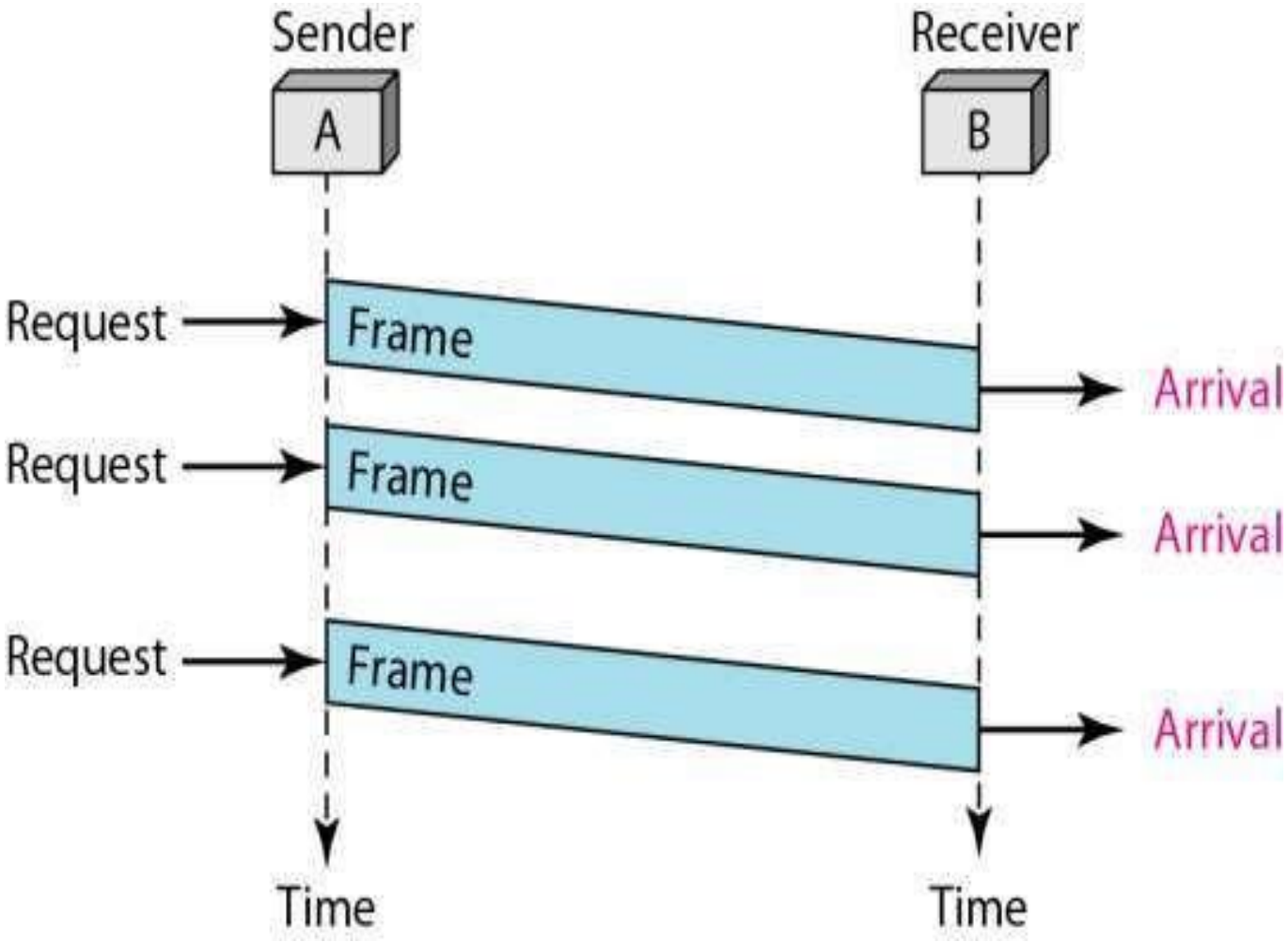
- The Simplex protocol is hypothetical protocol designed for unidirectional data transmission(i.e from sender to receiver) over an ideal channel, i.e. a channel through which transmission can never go wrong.

- In this protocol: Data are transmitted in one direction only
- The transmitting (Tx) and receiving (Rx) hosts are always ready
- Processing time can be ignored
- Infinite buffer space is available
- No sequence number or acknowledgements are used here.
- No errors occur; i.e. no damaged frames and no lost Frames



Simplest Protocol

Example : Simplest Protocol

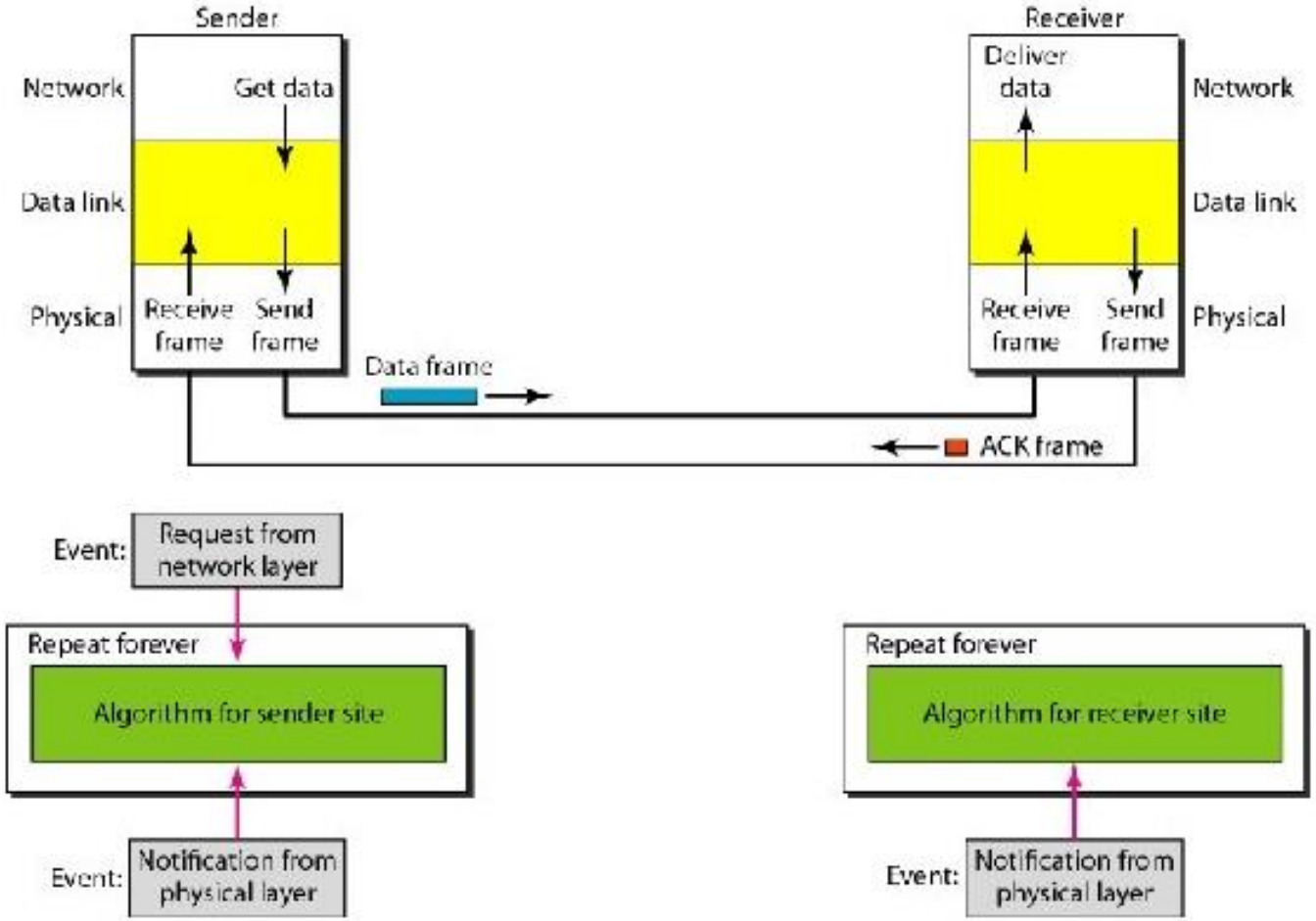


- This is unrealistic protocol ,because it does not handle either flow control or error correction

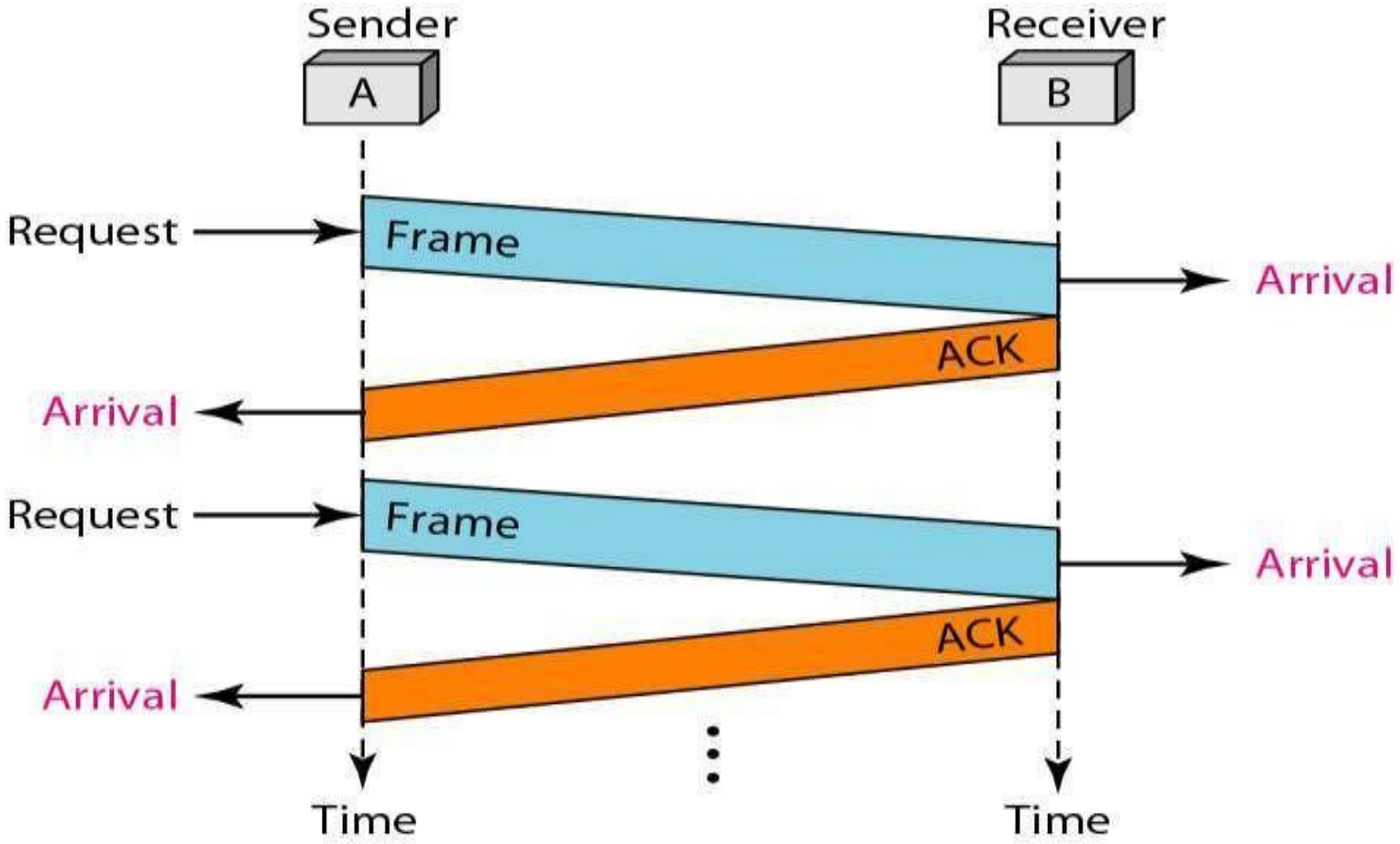
STOP & WAIT PROTOCOL

- The problem here is how to prevent the sender from flooding the receiver.
- Stop – and – Wait protocol is for noiseless channel too. It provides unidirectional data transmission without any error control facilities. However, it provides for flow control so that a fast sender does not drown a slow receiver
- The receiver send an acknowledge frame back to the sender telling the sender that the last received frame has been processed and passed to the host; permission to send the next frame is granted.

- The sender, after having sent a frame, must wait for the acknowledge frame from the receiver before sending another frame.
- This protocol is known as stop and wait protocol



Design of Stop and wait Protocol



Flow control for stop and wait

Drawbacks:

1.Only one frame can be in transmission at a time.

2.This leads to inefficiency if propagation delay is much longer than transmission delay.

Stop & Wait ARQ(Automatic Repeat Request)

- Stop & Wait ARQ is a **sliding window protocol** for flow control and it overcomes the limitations of Stop & Wait, we can say that it is the improved or modified version of Stop & Wait protocol.

- Working of Stop & Wait ARQ is almost like Stop & Wait protocol, the only difference is that it includes some additional components, which are:

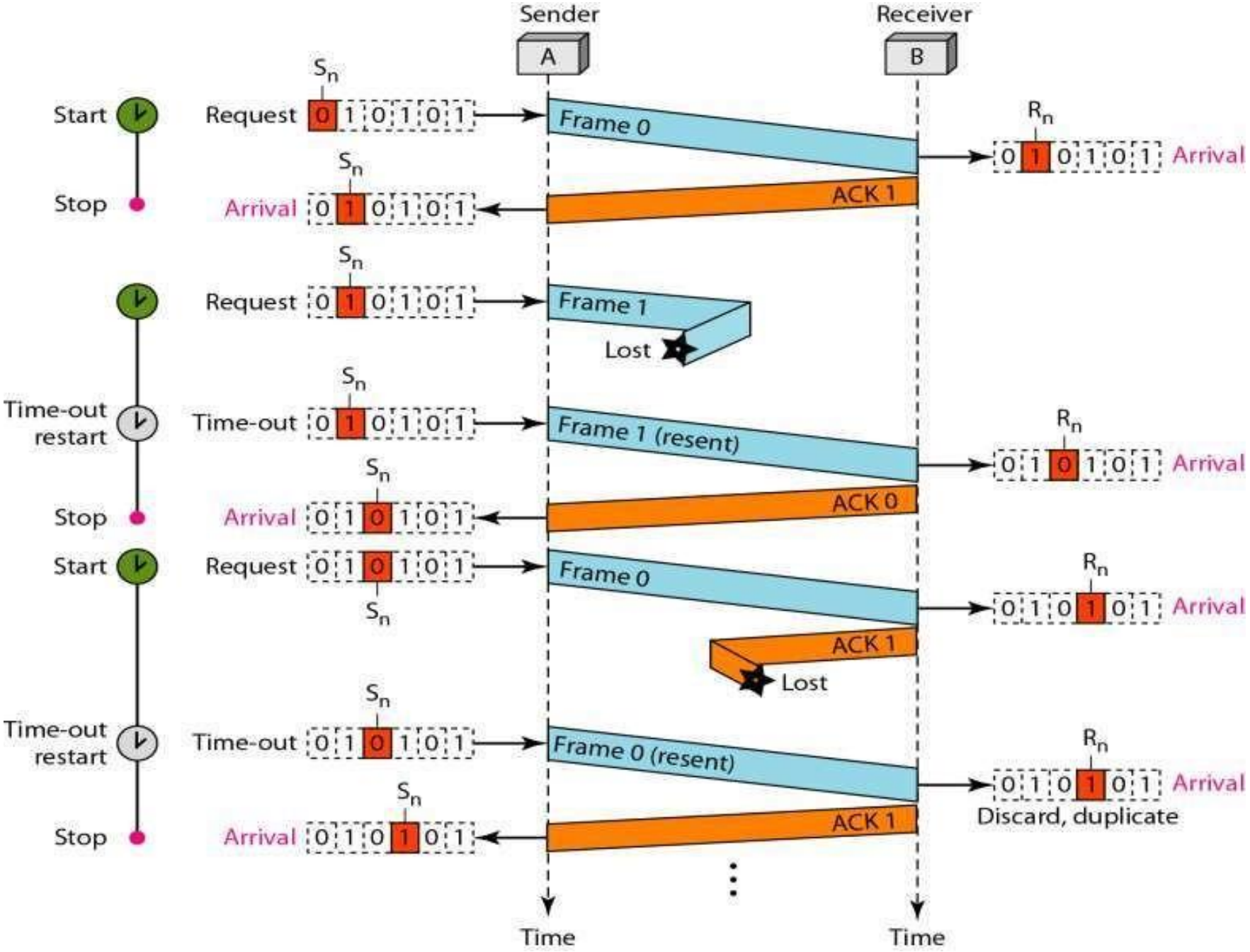
    1. Time out timer
    2. Sequence numbers for data packets
    3. Sequence numbers for feedbacks

- When the frame arrives at the receiver site, it is checked and if it is corrupted, it is silently discarded.
- Lost frames are more difficult to handle than corrupted ones. In our previous protocols, there was no way to identify a frame.
- When the receiver receives a data frame that is out of order, this means that frames were either lost or duplicated
- The received frame could be the correct one, or a duplicate, or a frame out of order. The solution is to number the frames.
- The lost frames need to be resent in this protocol. If the receiver does not respond when there is an error, how can the sender know which frame to resend?

- To remedy this problem, 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, the frame is resent, the copy is held, and the timer is restarted.
- Error correction in Stop-and-Wait ARQ is done by keeping a copy of the sent frame and retransmitting of the frame when the timer expires.

Operation:

- The sender transmits the frame, when frame arrives at the receiver it checks for damage and acknowledges to the sender accordingly. while transmitting a frame there can be 4 situations.
    1. Normal operation
    2. The frame is lost
    3. The acknowledgement is lost
    4. The acknowledgement is delayed


Sender
Receiver
A
B
Start
Stop
Request
Arrival
Frame 0
ACK 1
Frame 1
Lost
Time-out
restart
Time-out
Frame 1 (resent)
ACK 0
Frame 0
ACK 1
Lost
Frame 0 (resent)
Discard, duplicate
Time
Time

Limitation of Stop and Wait ARQ:

- The major limitation of Stop and Wait ARQ is its very less efficiency. To increase the efficiency, protocols like **Go back N** and **Selective Repeat** are used.

Stop and Wait Protocol Vs Stop and Wait ARQ

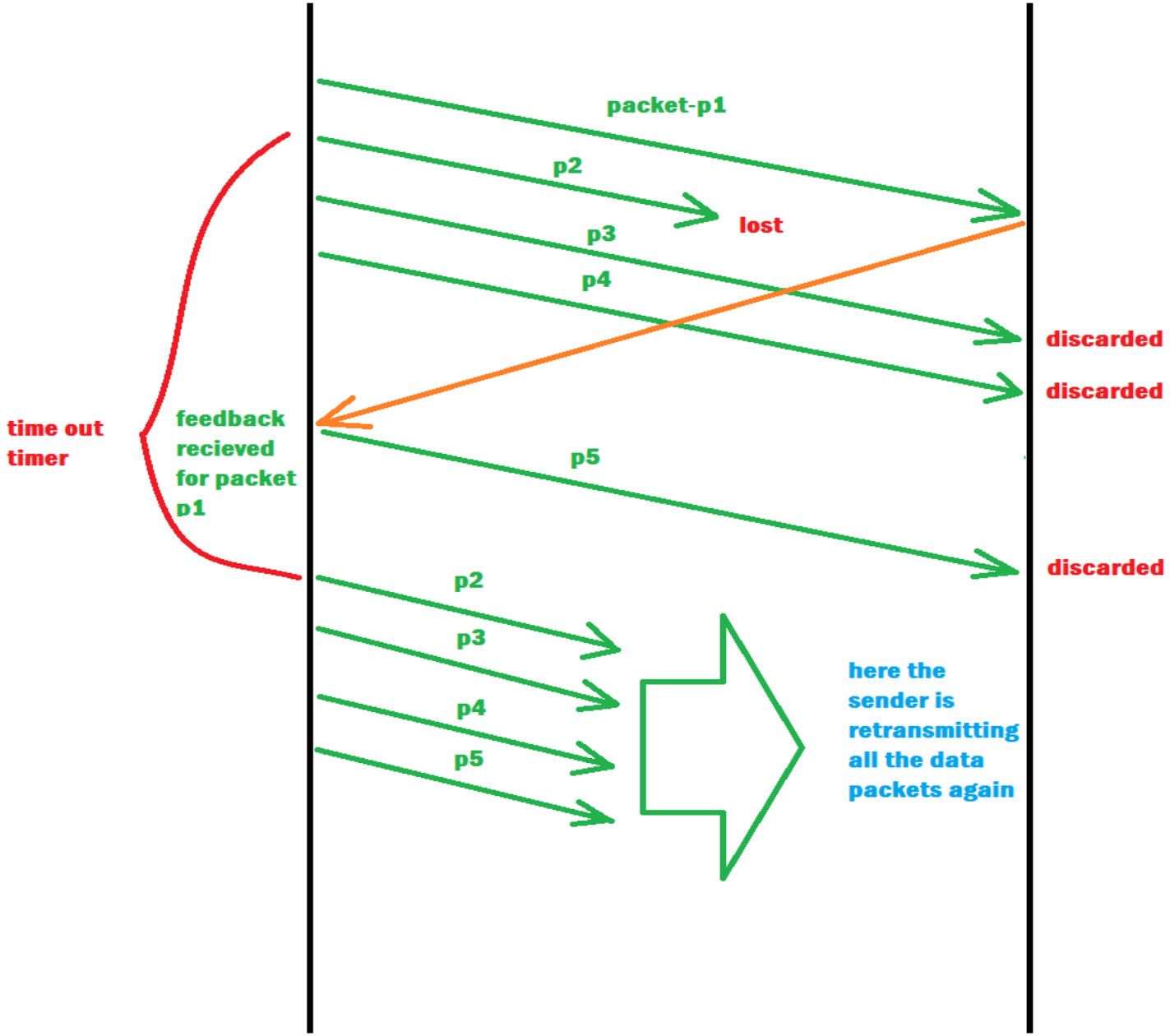
Stop and Wait Protocol	Stop and Wait ARQ
It assumes that the communication channel is perfect and noise free.	It assumes that the communication channel is imperfect and noisy.
Data packet sent by the sender can never get corrupt.	Data packet sent by the sender may get corrupt.
There is no concept of negative acknowledgements.	A negative acknowledgement is sent by the receiver if the data packet is found to be corrupt.
There is no concept of time out timer.	Sender starts the time out timer after sending the data packet.
There is no concept of sequence numbers.	Data packets and acknowledgements are numbered using sequence numbers.

Go-Back-N ARQ

Go-Back-N ARQ (Go-Back-N automatic repeat request) is a flow control protocol where the sender continues to send several frames specified by a window size even without receiving feedback from the receiver node.

Let's take an example

Consider a sender has to send data packets indexing from **p1 to p5**, it sends all the data packets in order (from p1 to p5), but the receiver has only received p1 and the data packet p2 is lost somewhere in the network, then the receiver declines all the data packets after p2 (i.e. p3, p4, p5) because the receiver is waiting for packet p2 and will not accept any other data packet than that. So, now as the time out time index of p2 expires, the sender goes back 3 packets and starts sending all the data packets from p2 to p5 again.


packet-p1
p2
lost
p3
p4
discarded
discarded
time out
timer
feedback
recieved
for packet
p1
p5
discarded
p2
p3
p4
p5
here the
sender is
retransmitting
all the data
packets again

Timer:

- Although there can be a timer for each frame that is sent, in our protocol we use only one.

Acknowledgment

- The receiver sends a positive acknowledgment if a frame has arrived safe .If a frame is damaged or is received out of order, the receiver is silent and will discard all subsequent frames until it receives the one it is expecting.

- The receiver does not have to acknowledge each frame received. It can send one cumulative acknowledgment for several frames.

Resending a Frame

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

- In Go-Back-N ARQ, The receiver keeps track of only one variable, and there is no need to buffer out-of- order frames; they are simply discarded. However, this protocol is very inefficient for a noisy link.

- In a noisy link a frame has a higher probability of damage, which means the resending of multiple frames. This resending uses up the bandwidth and slows down the transmission

- For noisy links, there is another mechanism that does not resend N frames when just one frame is damaged; only the damaged frame is resent. This mechanism is called Selective Repeat ARQ.

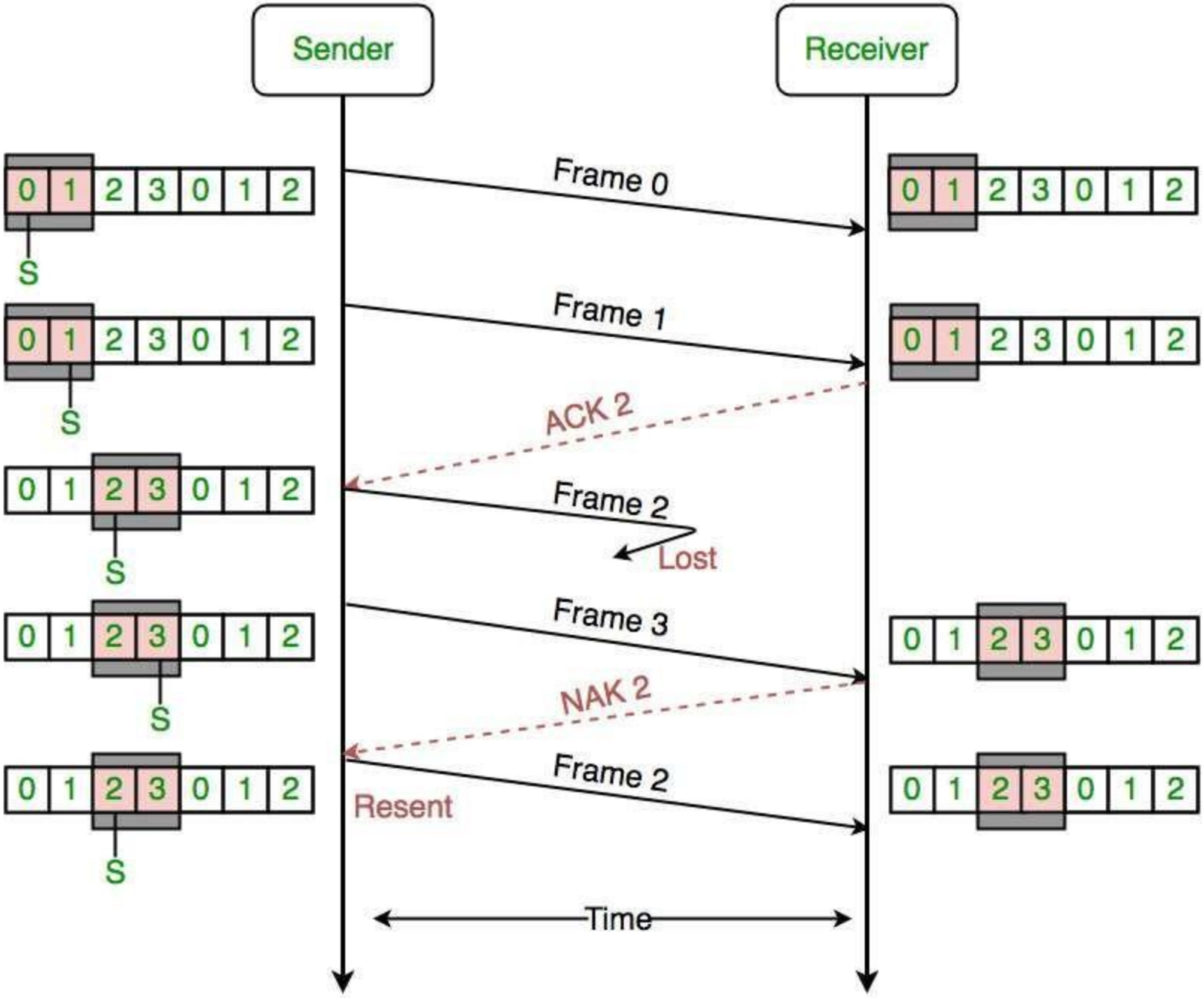
Selective Repeat Protocol (SRP) :

- This protocol(SRP) is mostly identical to GBN protocol, except that buffers are used and the receiver, and the sender, each maintain a window of size.

- SRP works better when the link is very unreliable.

- Because in this case, retransmission tends to happen more frequently, selectively retransmitting frames is more efficient than retransmitting all of them.

- SRP also requires full duplex link. backward acknowledgements are also in progress.

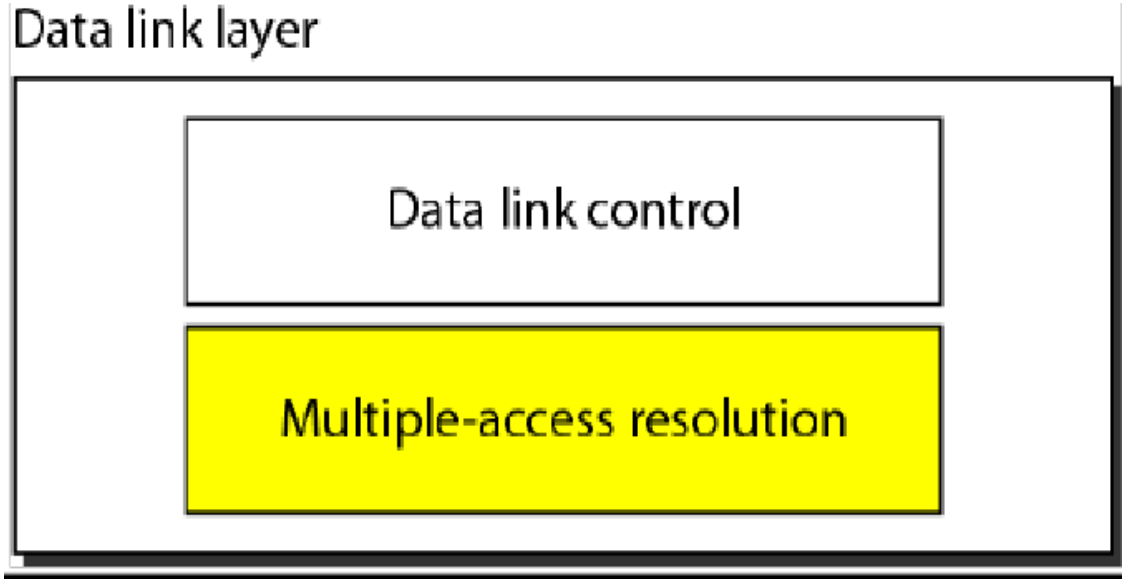

Sender
Receiver
0 1 2 3 0 1 2
S
Frame 0
Frame 1
ACK 2
Frame 2
Lost
Frame 3
NAK 2
Frame 2
Resent
Time

UNIT II

MULTIPLE ACCESS PROTOCOLS

MULTIPLE ACCESS PROTOCOLS

- We can consider the data link layer as two sub layers. The upper sub layer is responsible for data link control, and the lower sub layer is responsible for resolving access to the shared media

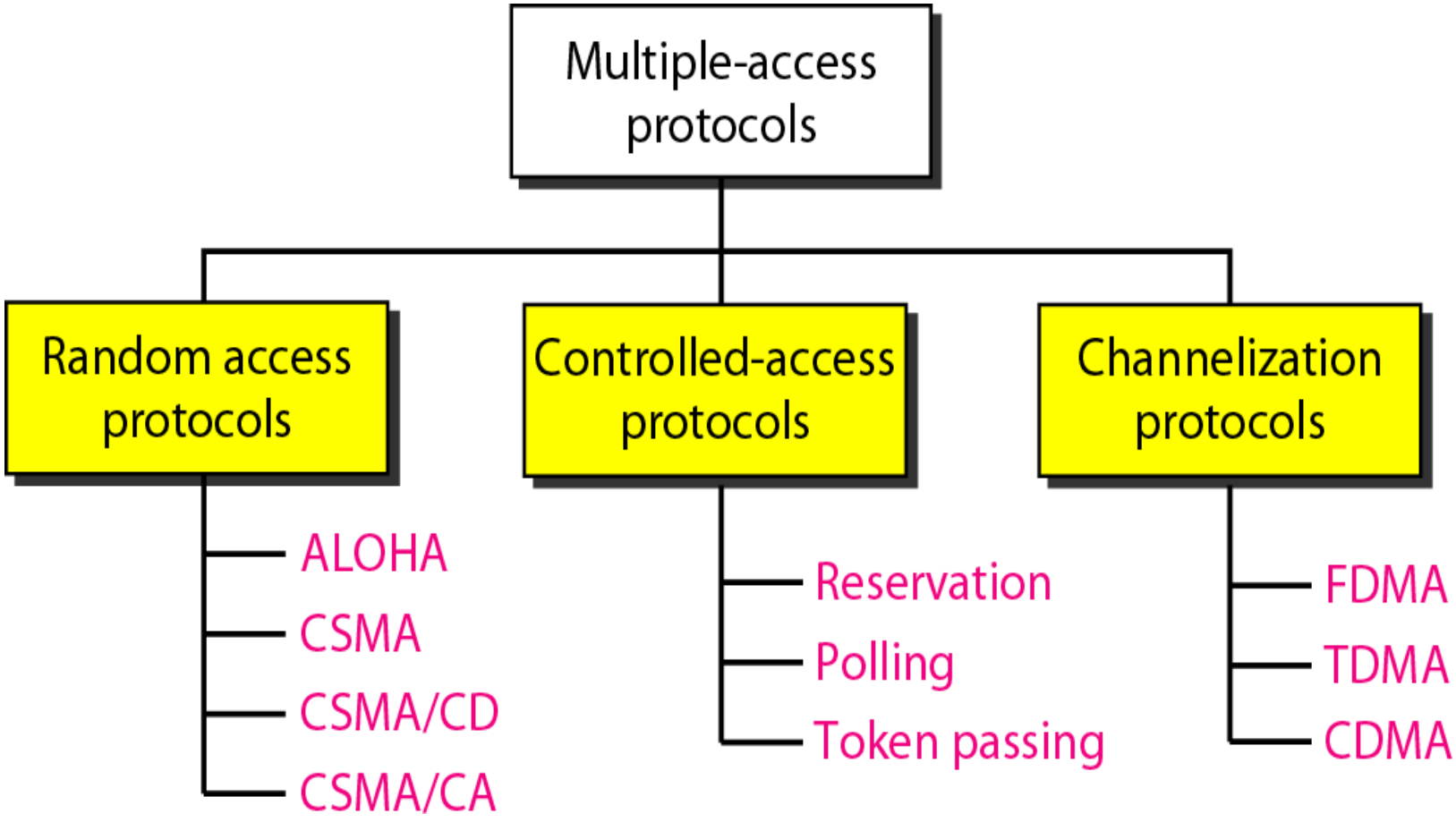


MULTIPLE ACCESS PROTOCOLS

- The upper sub layer that is responsible for flow and error control is called the logical link control (LLC) layer; the lower sub layer that is mostly responsible for multiple access resolution is called the media access control (MAC) layer.

- When nodes or stations are connected and use a common link, called a multipoint or broadcast link, we need a multiple-access protocol to coordinate access to the link.

MULTIPLE ACCESS PROTOCOLS



RANDOM ACCESS PROTOCOLS

- In random access or contention methods, no station is superior to another station and none is assigned the control over another.
- Two features give this method its name.
- First, 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.
- Second, no rules 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.

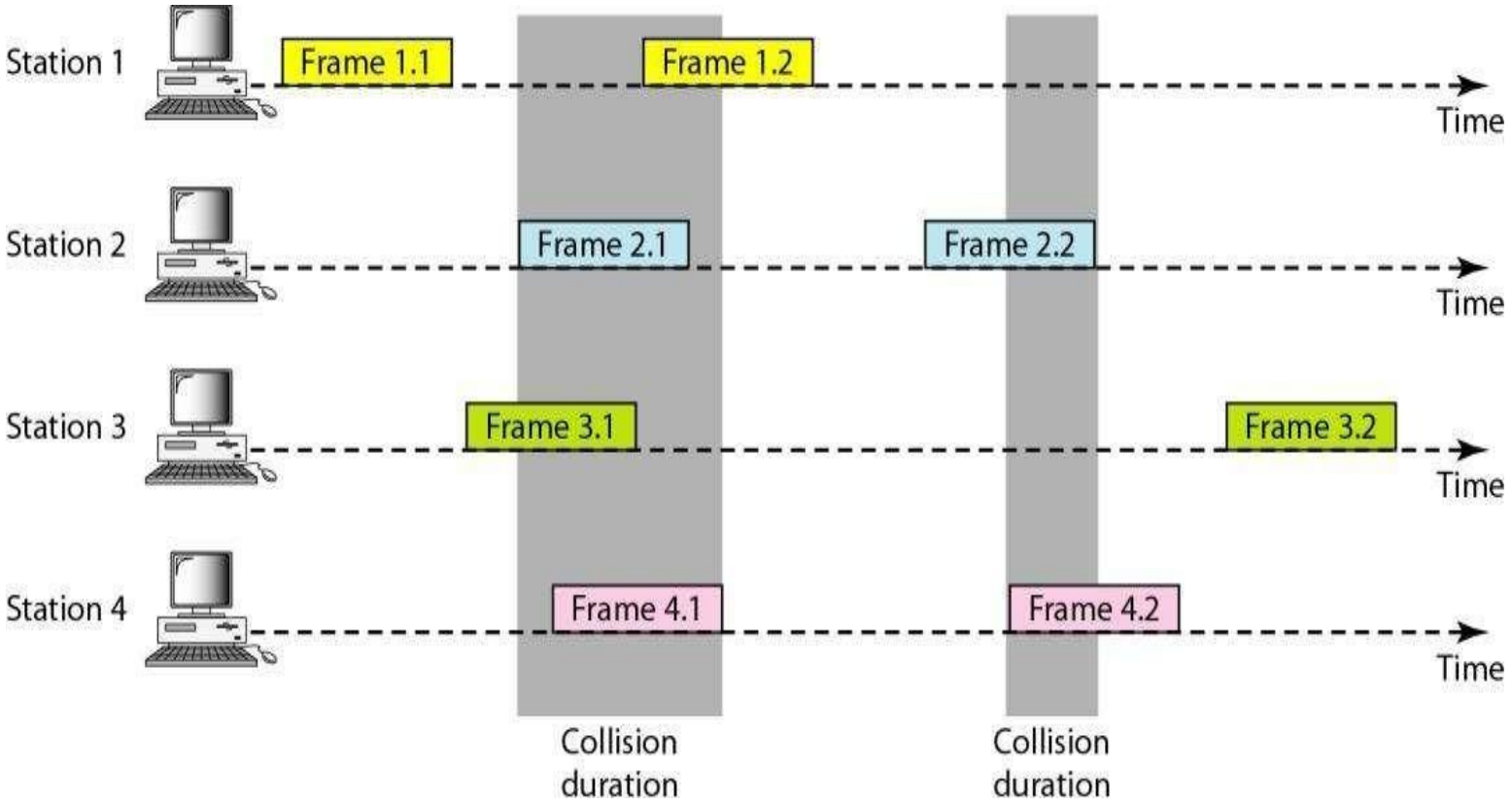
ALOHA

1. Pure ALOHA

- The original ALOHA protocol is called pure ALOHA. This is a simple, but elegant protocol.

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

- Below Figure shows an example of frame collisions in pure ALOHA.



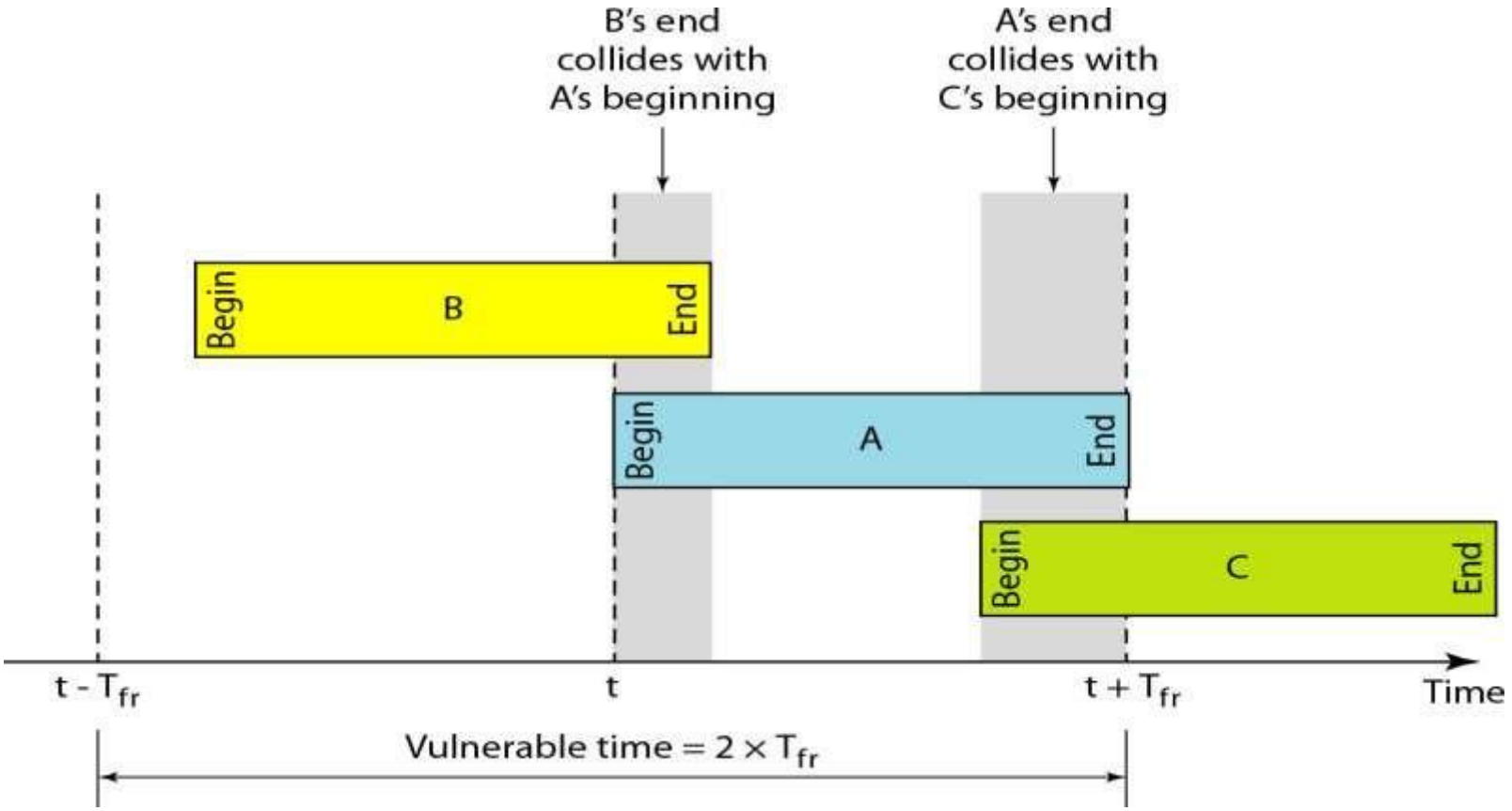
Frames in a pure ALOHA network

- In pure ALOHA, whenever any station transmits a frame, it expects the acknowledgement from the receiver. If acknowledgement is not received within specified time, the station assumes that the frame (or acknowledgement) has been destroyed.
- If the frame is destroyed because of collision the station waits for a random amount of time and sends it again.
- This waiting time must be random otherwise same frames will collide again and again. This randomness will help avoid more collisions.

Vulnerable time

- Let us find the length of time, the vulnerable time, in which there is a possibility of collision.

We assume that the stations send fixed- length frames with each frame taking Tfr S to send. Below Figure shows the vulnerable time for station A.



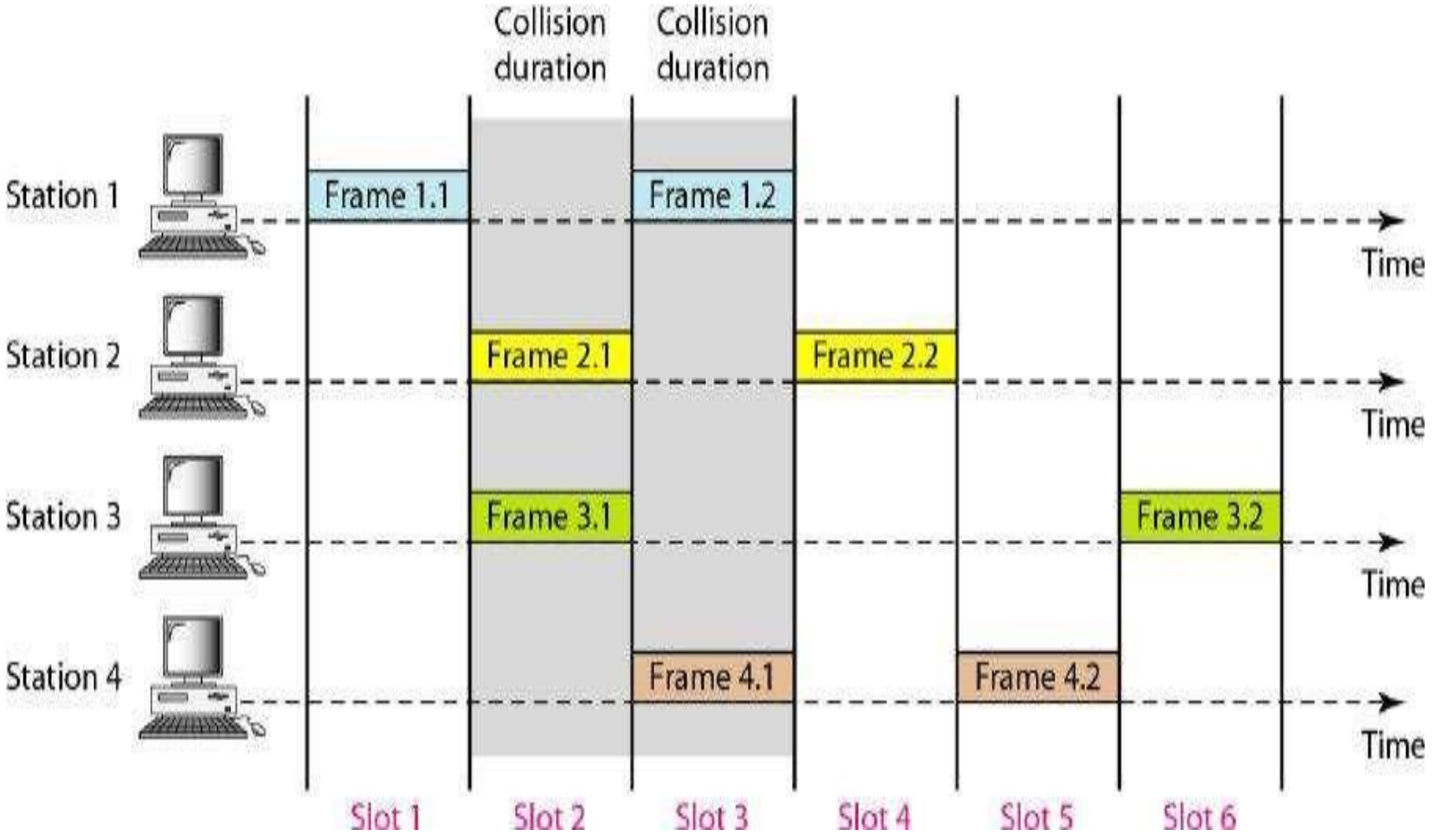
The throughput for pure ALOHA is $S = G \times e^{-2G}$. The maximum throughput Smax = 0.184 when G= (1/2).

- Station A sends a frame at time t. Now imagine station B has already sent a frame between t - Tfr 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. On the other hand, suppose that station C sends a frame between t and t + Tfr .
- 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
- Looking at Figure, we see that 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 Tfr

2.Slotted ALOHA

•Pure ALOHA has a vulnerable time of 2 x Tfr . 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 Tfr s and force the station to send only at the beginning of the time slot. Figure 3 shows an example of frame collisions in slotted ALOHA



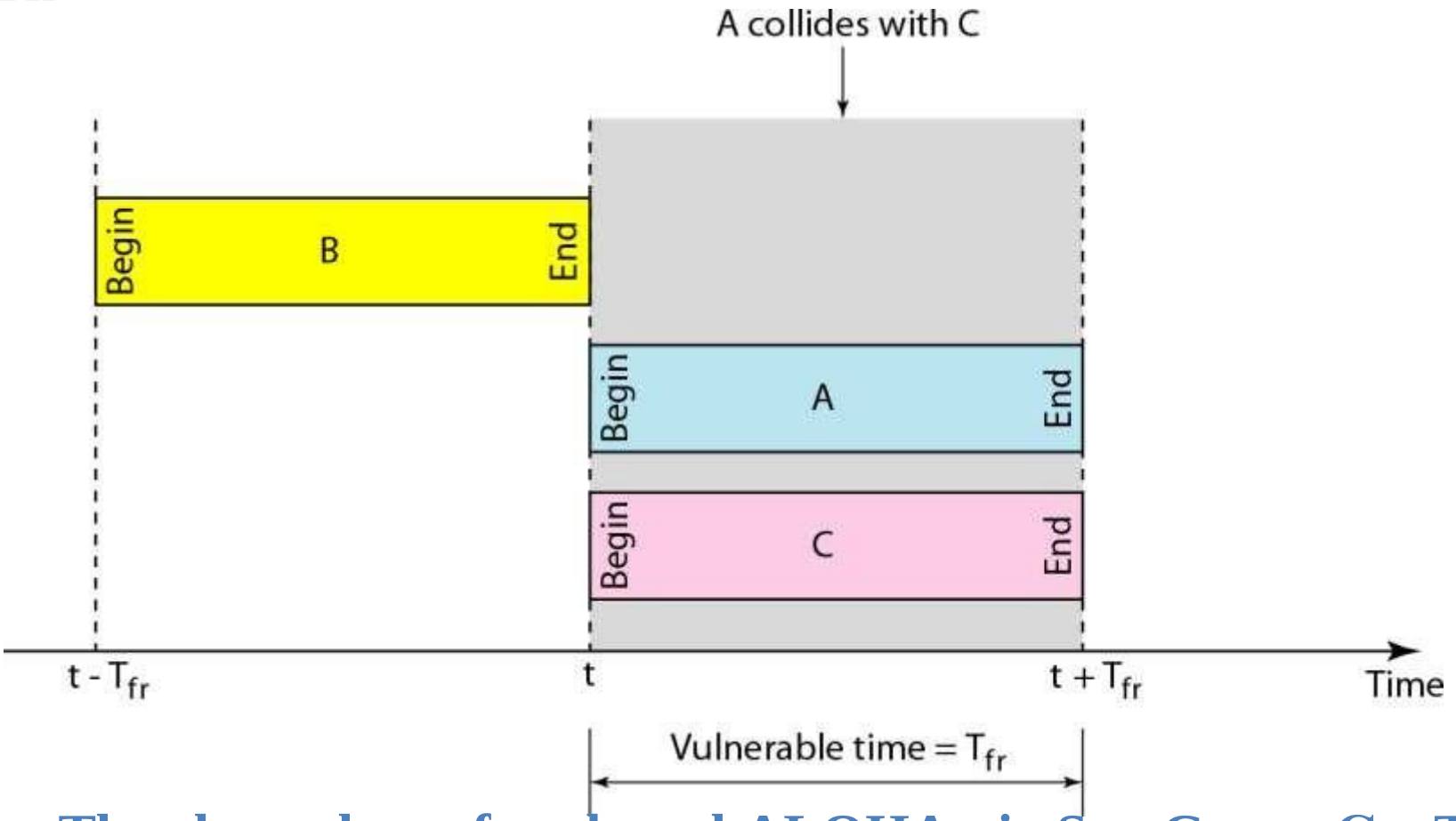
Collisions in slotted ALOHA

- Because 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. Of course, there is still the possibility of collision if two stations try to send at the beginning of the same time slot.

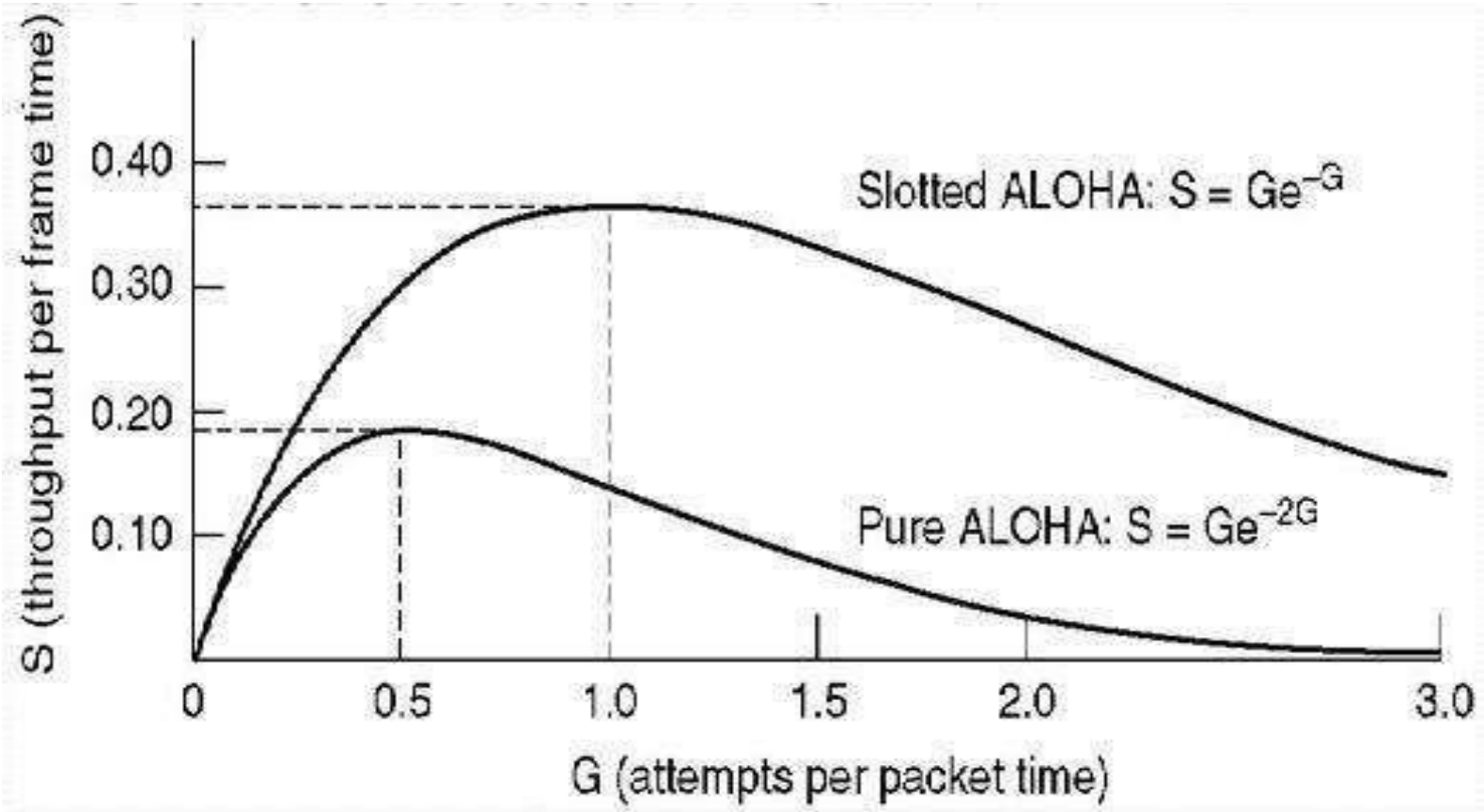
- However, the vulnerable time is now reduced to one-half, equal to Tfr . Below figure shows the situation

- Below fig shows that the vulnerable time for slotted ALOHA is one-half that of pure ALOHA. Slotted ALOHA vulnerable time = Tfr



The throughput for slotted ALOHA is S = G × e−G . The maximum throughput Smax = 0.368 when G = 1.

Comparison between Pure Aloha & Slotted Aloha



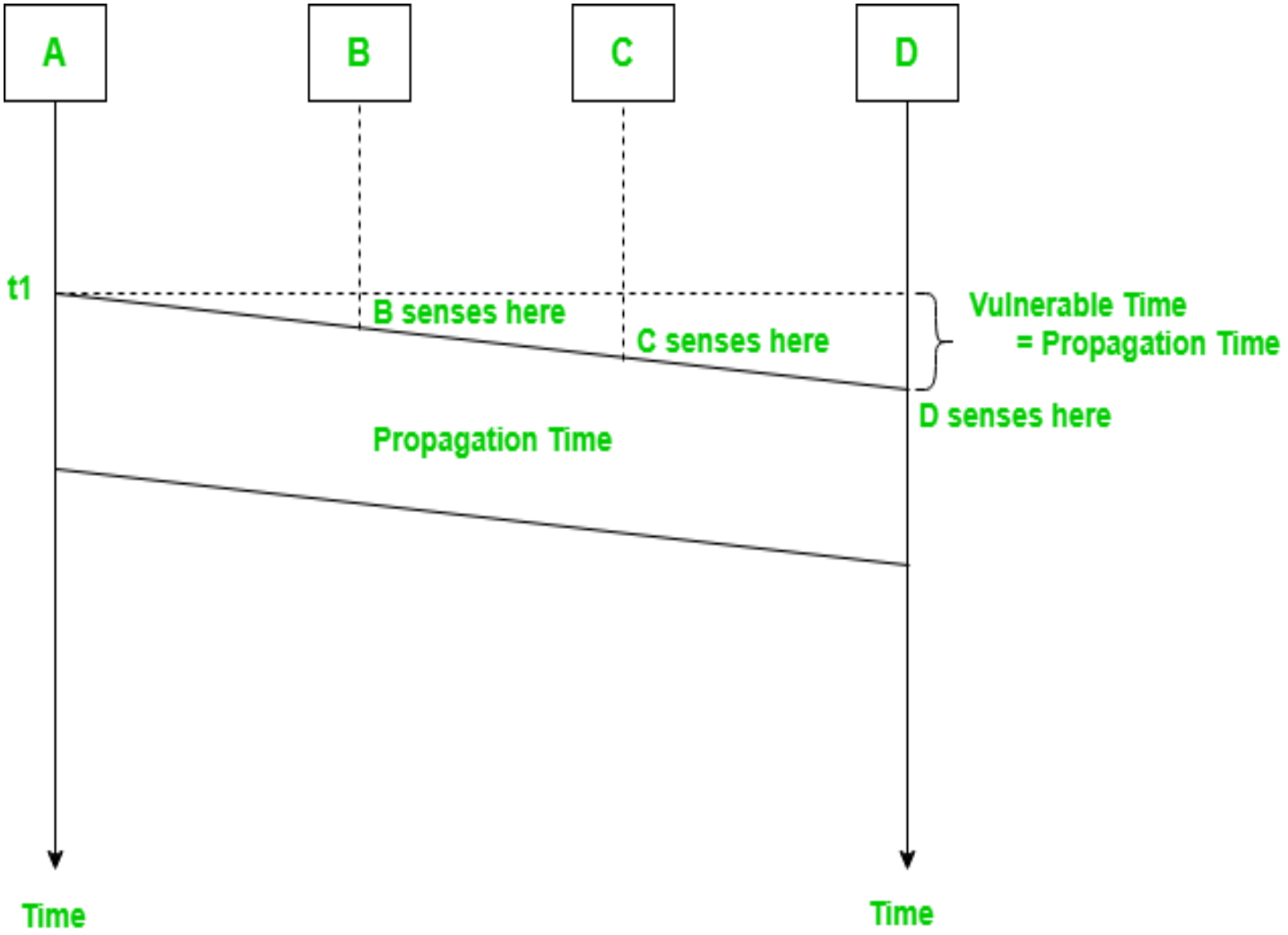
CarrierSenseMultipleAccess(CSMA)

- To minimize the chance of collision and, therefore, increase the performance, the CSMA method was developed.
- The chance of collision can be reduced if a station senses the medium before trying to use it.
- Carrier sense multiple access (CSMA) requires that each station first listen to the medium (or check the state of the medium) before sending.
- In other words, CSMA is based on the principle "sense before transmit" or "listen before talk."

- CSMA can reduce the possibility of collision, but it cannot eliminate it. The reason for this is shown in below Figure.

- The possibility of collision still exists because of propagation delay; station may sense the medium and find it idle, only because the first bit sent by another station has not yet been received.

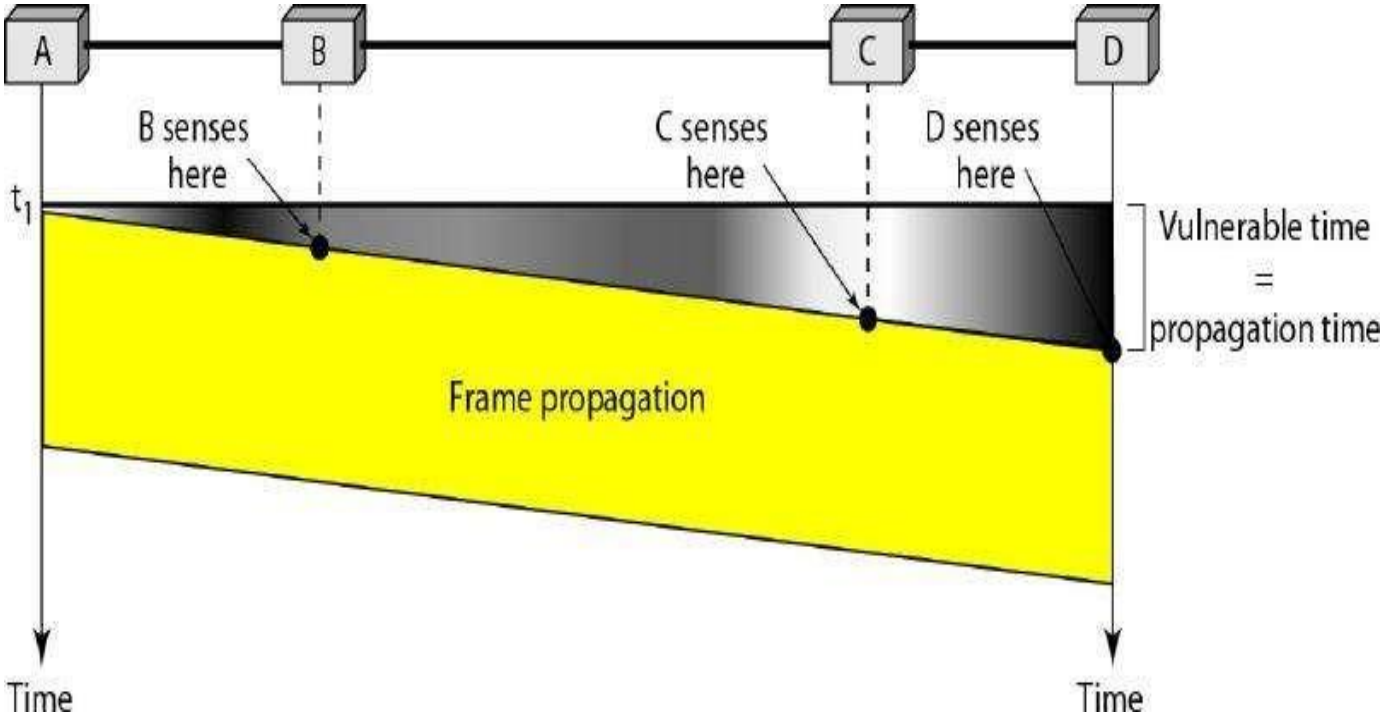
- At time tI' station B senses the medium and finds it idle, so it sends a frame. At time t2 (t2> tI)' station C senses the medium and finds it idle because, at this time, the first bits from station B have not reached station C. Station C also sends a frame. The two signals collide and both frames are destroyed.



Space/time model of the collision in CSMA

Vulnerable Time

•The vulnerable time for CSMA is the propagation time Tp. This is the time needed for a signal to propagate from one end of the medium to the other.



Persistence Methods

- What should a station do if the channel is busy? What should a station do if the channel is idle? Three methods have been devised to answer these questions:

- **1-persistent method**
- **non-persistent method**
- **p-persistent method**

- **1-Persistent:** 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.

- **Non-persistent:** 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. This approach reduces the chance of collision However, this method reduces the efficiency of the network because the medium remains idle when there may be stations with frames to send.

p-Persistent: This is used if the channel has time slots with a slot duration equal to or greater than the maximum propagation time. The p-persistent approach combines the advantages of the other two strategies. It reduces the chance of collision and improves efficiency.

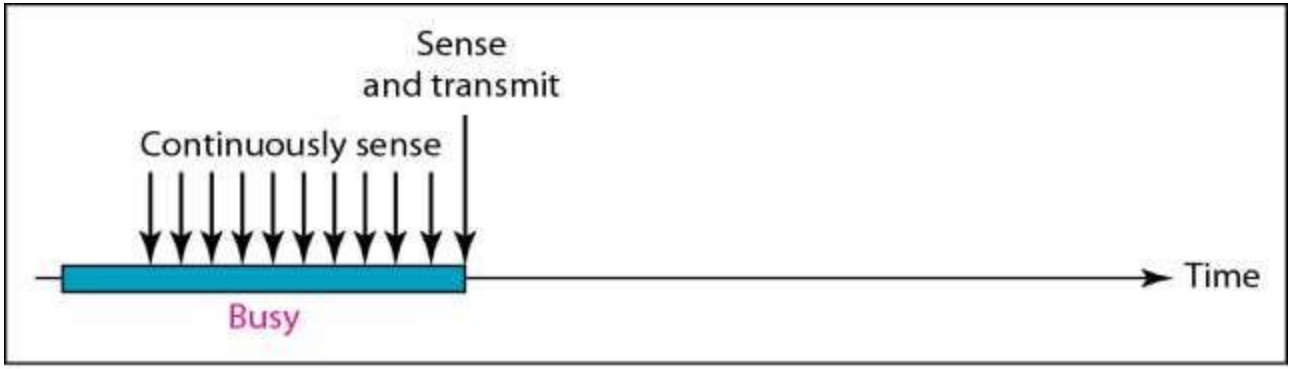
In this method, after the station finds the line idle it follows these steps:

1. With probability p, the station sends its frame.

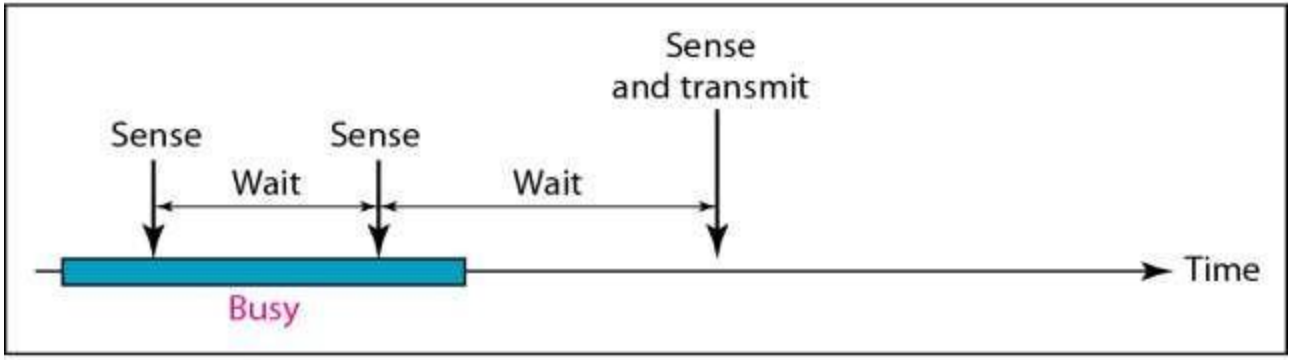
2. With probability q = 1-p, the station waits for the beginning of the next time slot and checks the line again.

a. If the line is idle, it goes to step 1.

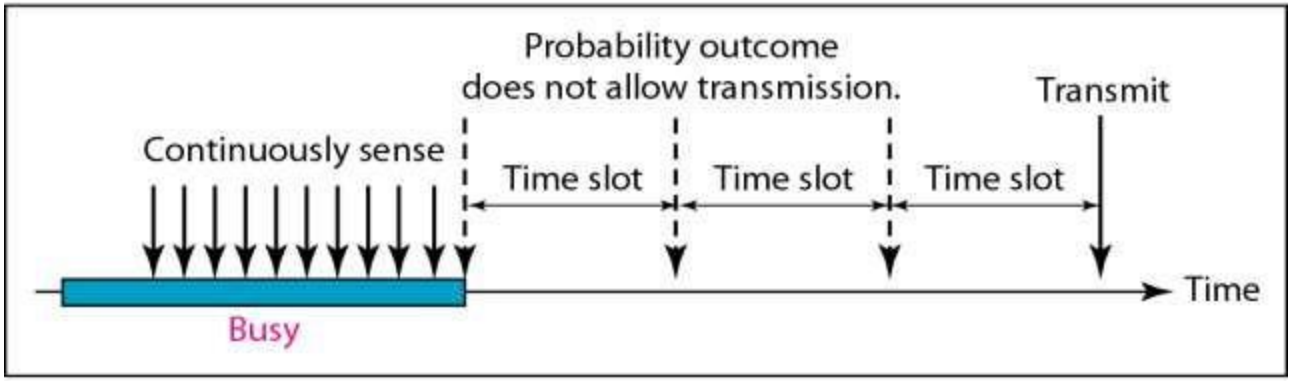
b. If the line is busy, it acts as though a collision has occurred and uses the backoff procedure.



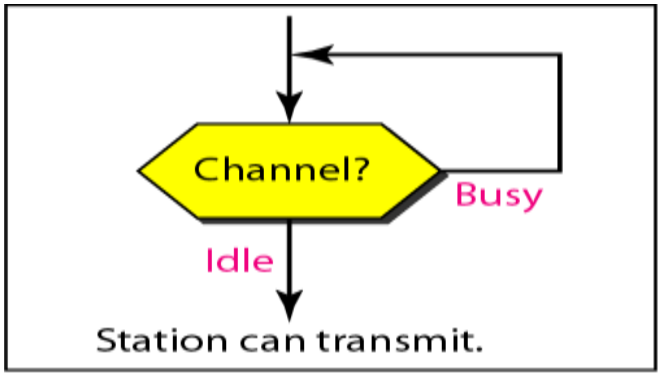
a. 1-persistent



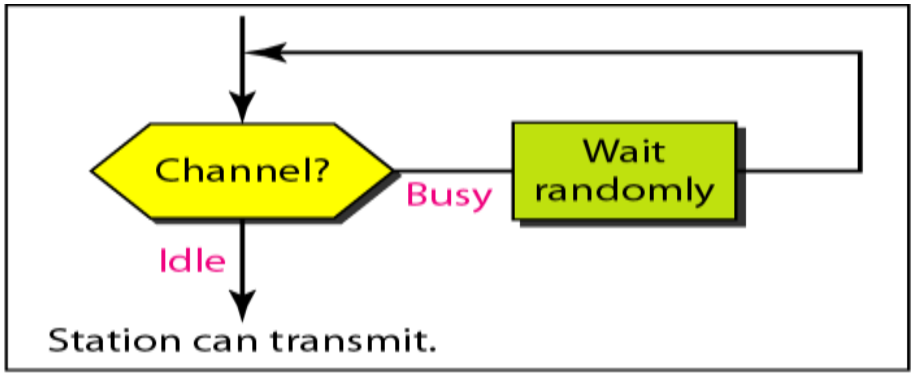
b. Nonpersistent



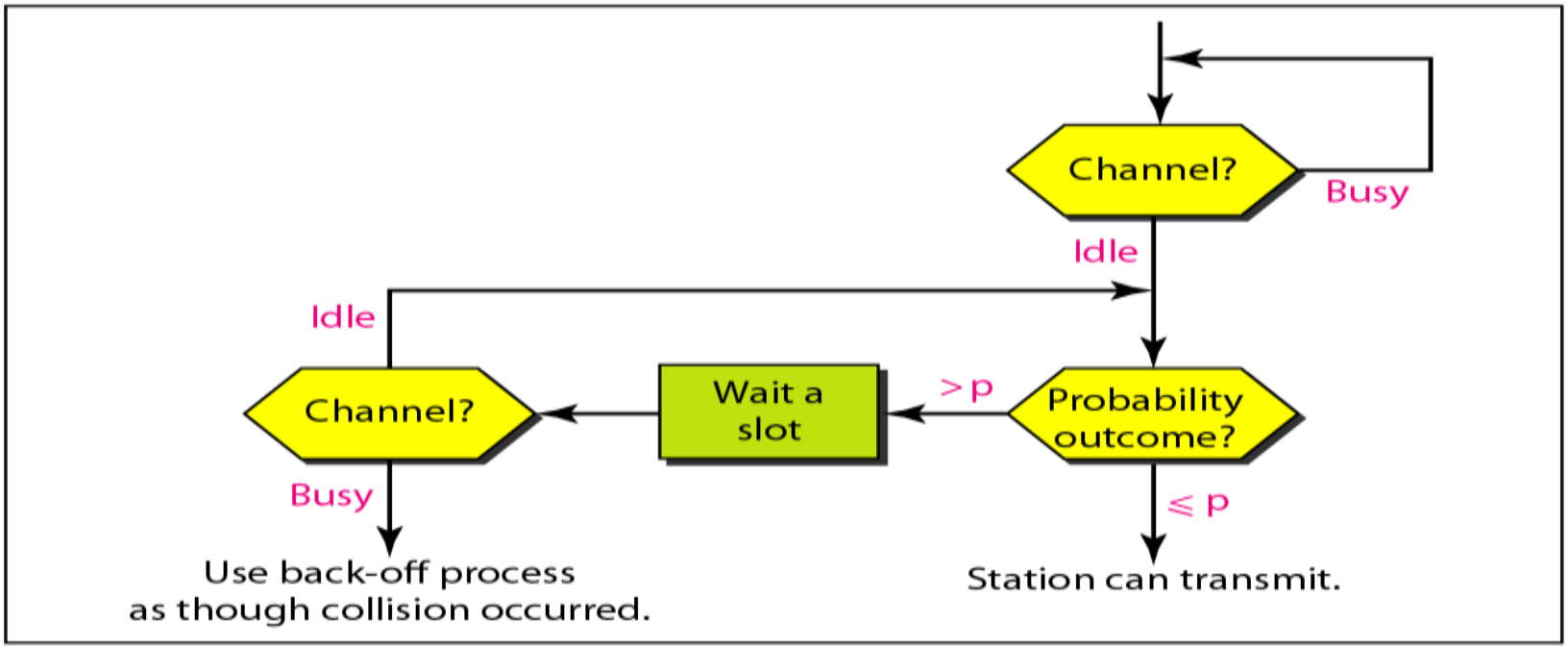
c. p-persistent



a. 1-persistent



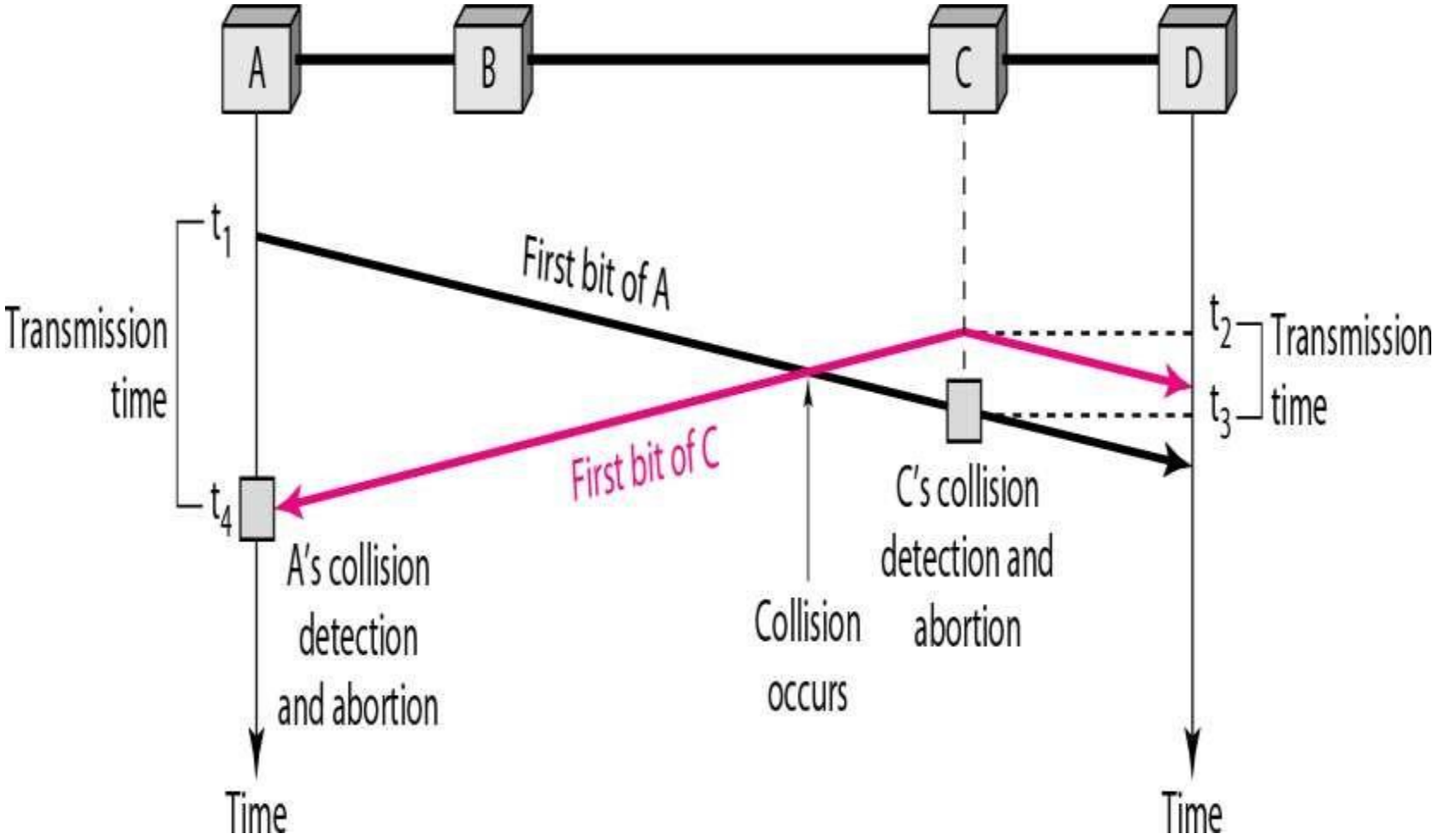
b. Nonpersistent



c. p-persistent

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

- The CSMA method does not specify the procedure following a collision. 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 successful, the station is finished, if not, the frame is sent again.
- To better understand CSMA/CD, let us look at the first bits transmitted by the two stations involved in the collision. Although each station continues to send bits in the frame until it detects the collision, we show what happens as the first bits collide. In below Figure, stations A and C are involved in the collision.



Collision of the first bit in CSMA/CD

How CSMA/CD works?

- **Step 1**: Check if the sender is ready for transmitting data packets.

- **Step 2:** Check if the transmission link is idle? Sender has to keep on checking if the transmission link/medium is idle. For this it continuously senses transmissions from other nodes. Sender sends dummy data on the link. If it does not receive any collision signal, this means the link is idle at the moment. If it senses that the carrier is free and there are no collisions, it sends the data. Otherwise it refrains from sending data.

•**Step 3:** Transmit the data & check for collisions.
Sender transmits its data on the link. CSMA/CD does not use 'acknowledgement' system.

•It checks for the successful and unsuccessful transmissions through collision signals. During transmission, if collision signal is received by the node, transmission is stopped.

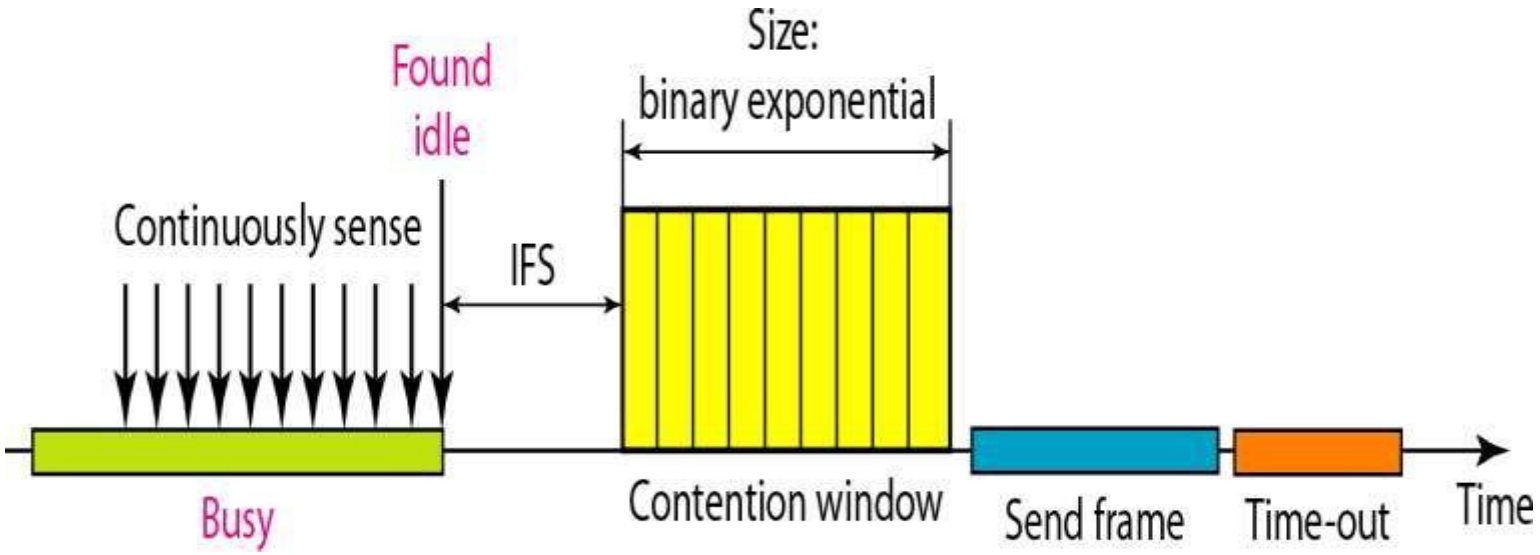
•The station then transmits a jam signal onto the link and waits for random time interval before it resends the frame. After some random time, it again attempts to transfer the data and repeats above process.

•**Step 4:** If no collision was detected in propagation, the sender completes its frame transmission and resets the counters.

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

- We need to avoid collisions on wireless networks because they cannot be detected. Carrier sense multiple access with collision avoidance (CSMA/CA) was invented for wirelesss network. Collisions are avoided through the use of CSMA/CA's three strategies:

- the inter frame space(IFS),
- the contention window, and
- acknowledgments, as shown in Below Figure


Found
idle
Size:
binary exponential
Continuously sense
IFS
Busy
Contention window
Send frame
Time-out
Time

- **Interframe space:** in this case, assume that your station waits for the channel to become idle and found that the channel is idle, then it will not send the data-frame immediately (in order to avoid collision due to propagation delay) it rather waits for some time called interframe space or IFS, and after this time the station again checks the medium for being idle. But it should be kept in mind that the IFS duration depends on the priority of the station.

- **Contention Window**: here, the time is divided into slots. Say, if the sender is ready for transmission of the data, it then chooses a random number of slots as waiting time which doubles every time whenever the channel is busy. But, if the channel is not idle at that moment, then it does not restart the entire process but restarts the timer when the channel is found idle again.

- **Acknowledgment:** as we discussed above that the sender station will re-transmits the data if acknowledgment is not received before the timer expires.

Collision-Free Protocols

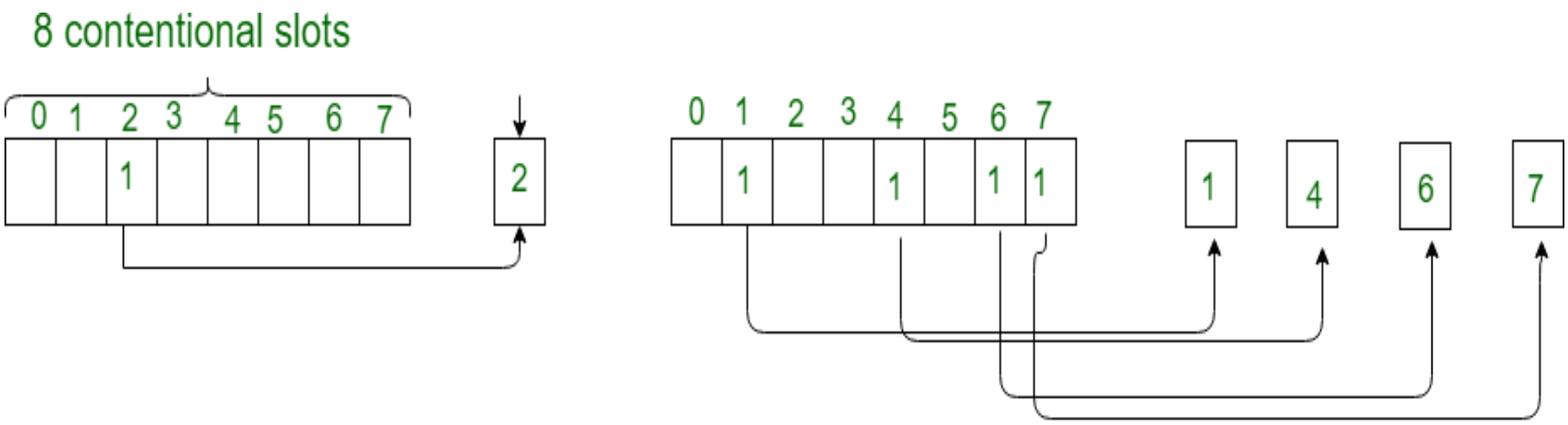
- Bit-map Protocol
- Binary Countdown
- The Adaptive Tree Walk Protocol

1. Bit-map Protocol:

- Bit map protocol is collision free Protocol . In bitmap protocol method, each contention period consists of exactly N slots.

- where N is the total number of stations sharing the channel. If a station has a frame to send, , then it transmits a 1 bit in the respective slot.

- For example if station 2 has a frame to send, it transmits a 1 bit during the second slot.

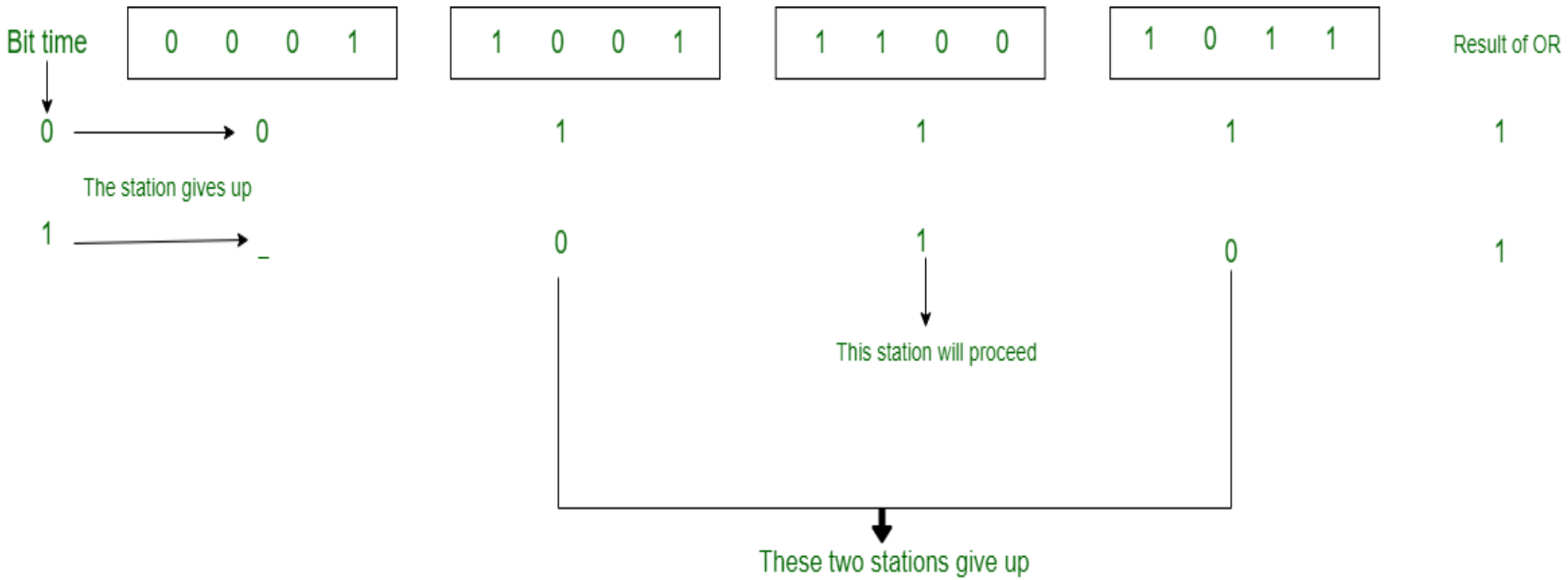


A Bit-map Protocol.

2. Binary Countdown:

- In binary countdown, binary station addresses are used. A station wanting to use the channel broadcast its address as binary bit string starting with the high order bit.
- All addresses are assumed of the same length. Here, we will see the example to illustrate the working of the binary countdown.
- For example, if there are 6 stations, they may be assigned the binary addresses 001, 010, 011, 100, 101 and 110. All stations wanting to communicate broadcast their addresses. The station with higher address gets the higher priority for transmitting.

- In this method, different station addresses are ORed together who decide the priority of transmitting. If these stations 0001, 1001, 1100, 1011 all are trying to seize the channel for transmission.
- All the station at first broadcast their most significant address bit that is 0, 1, 1, 1 respectively. The most significant bits are ORed together. Station 0001 see the 1MSB in another station addresses and knows that a higher numbered station is competing for the channel, so it gives up for the current round.
- Other three stations 1001, 1100, 1011 continue. The next bit is 1 at station 1100, swiss station 1011 and 1001 give up. Then station 110 starts transmitting a frame, after which another bidding cycle starts.

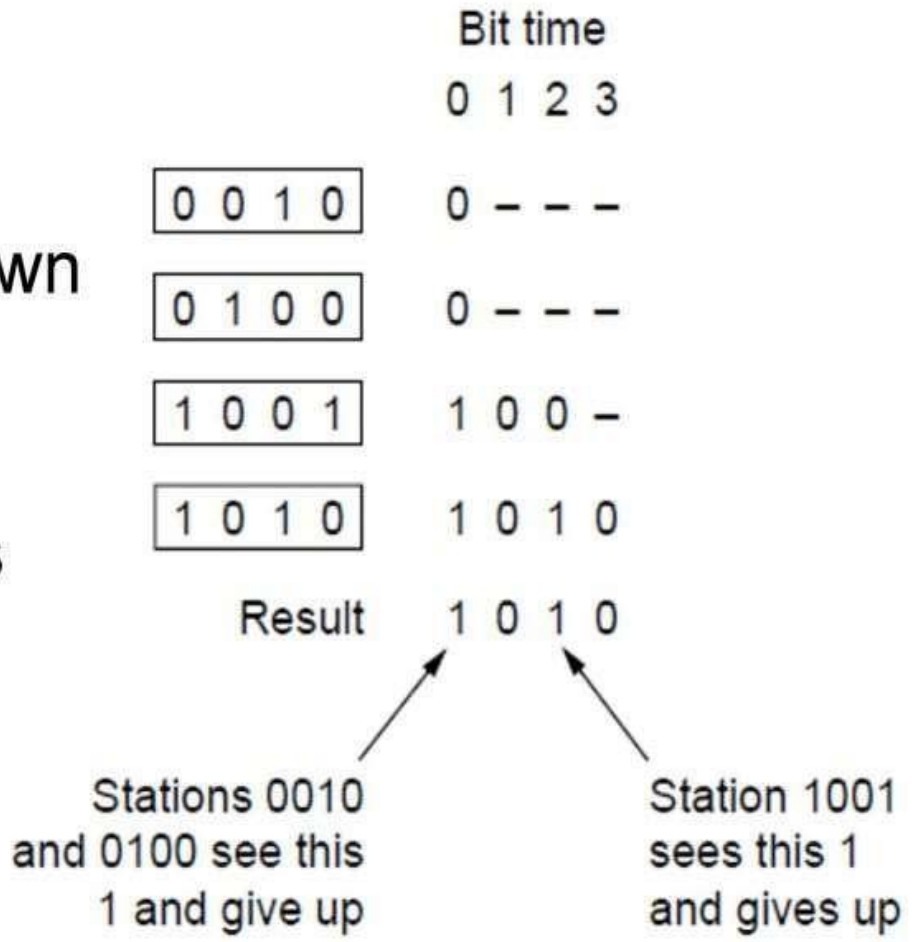


Binary countdown

Binary Countdown

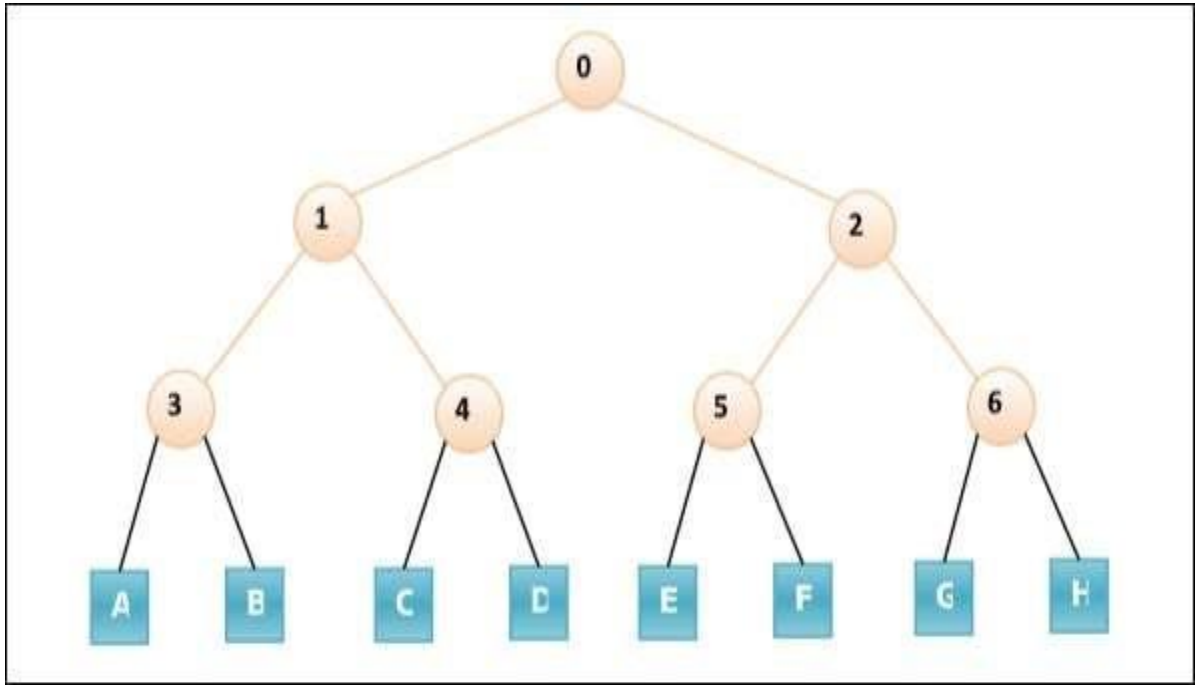
The binary countdown protocol.

A dash indicates silence.



Adaptive Tree Walk Protocol

- In adaptive tree walk protocol, the stations or nodes are arranged in the form of a binary tree as follows -



- Initially all nodes (A, B ……. G, H) are permitted to compete for the channel.
- If a node is successful in acquiring the channel, it transmits its frame.
- In case of collision, the nodes are divided into two groups (A, B, C, D in one group and E, F, G, H in another group).
- Nodes belonging to only one of them is permitted for competing. This process continues until successful transmission occurs.

Data Communications and Networking Fourth Edition

Forouzan

Wired LANs: Ethernet

IEEE STANDARDS

- In 1985, the Computer Society of the IEEE started a project, called Project 802, to set standards to enable intercommunication among equipment from a variety of manufacturers.

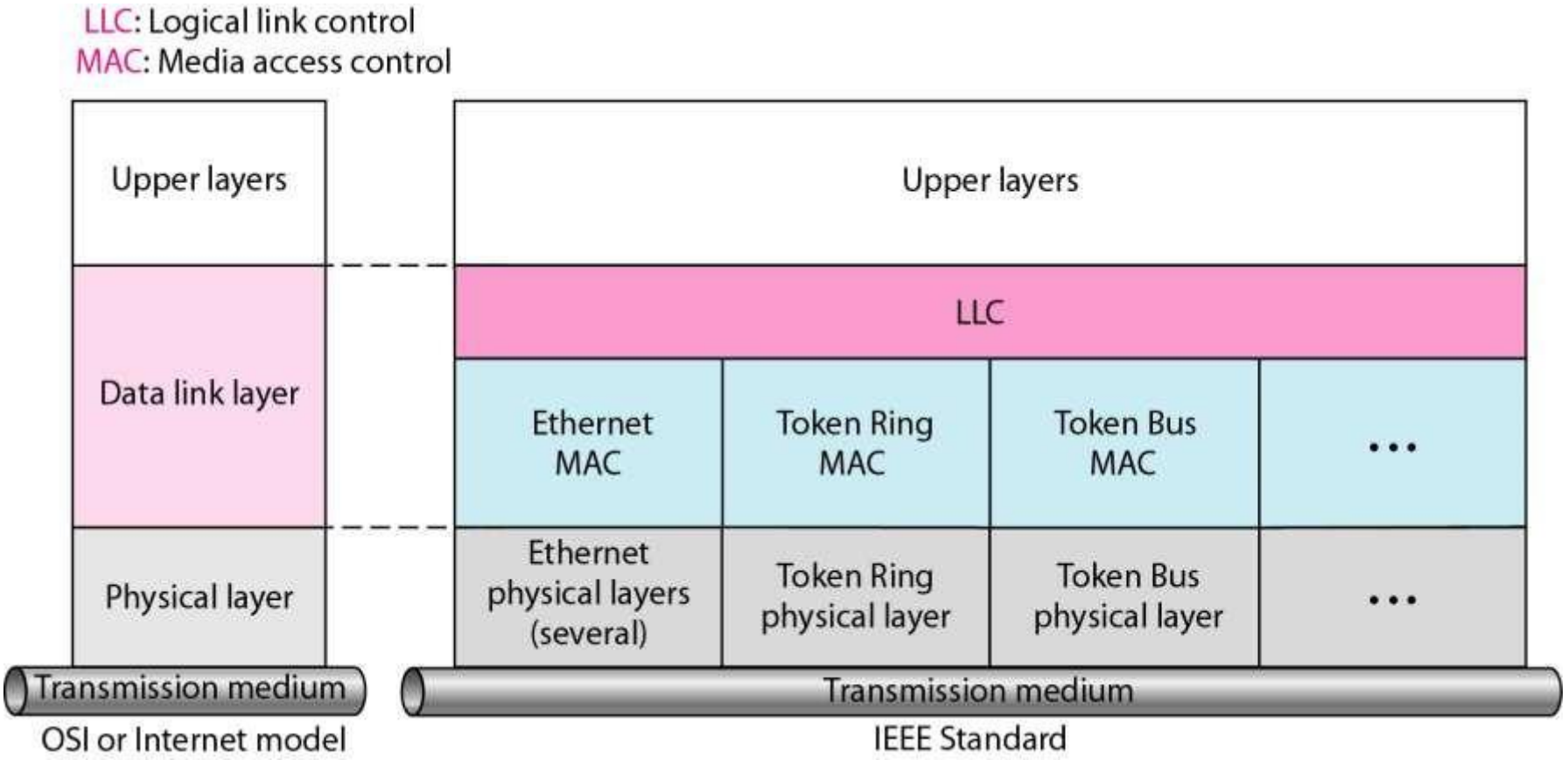
- Project 802 is a way of specifying functions of the physical layer and the data link layer of major LAN protocols.

- The relationship of the 802 Standard to the traditional OSI model is shown in below Figure.

- The IEEE has subdivided the data link layer into two sub layers: logical link control (LLC) and media access control(MAC).

Figure 13.1 *IEEE standard for LANs*

IEEE has also created several physical layer standards for different LAN protocols



STANDARD ETHERNET

- The original Ethernet was created in 1976 at Xerox's Palo Alto Research Center (PARC). Since then, it has gone through four generations. We briefly discuss the Standard (or traditional) Ethernet in this section.

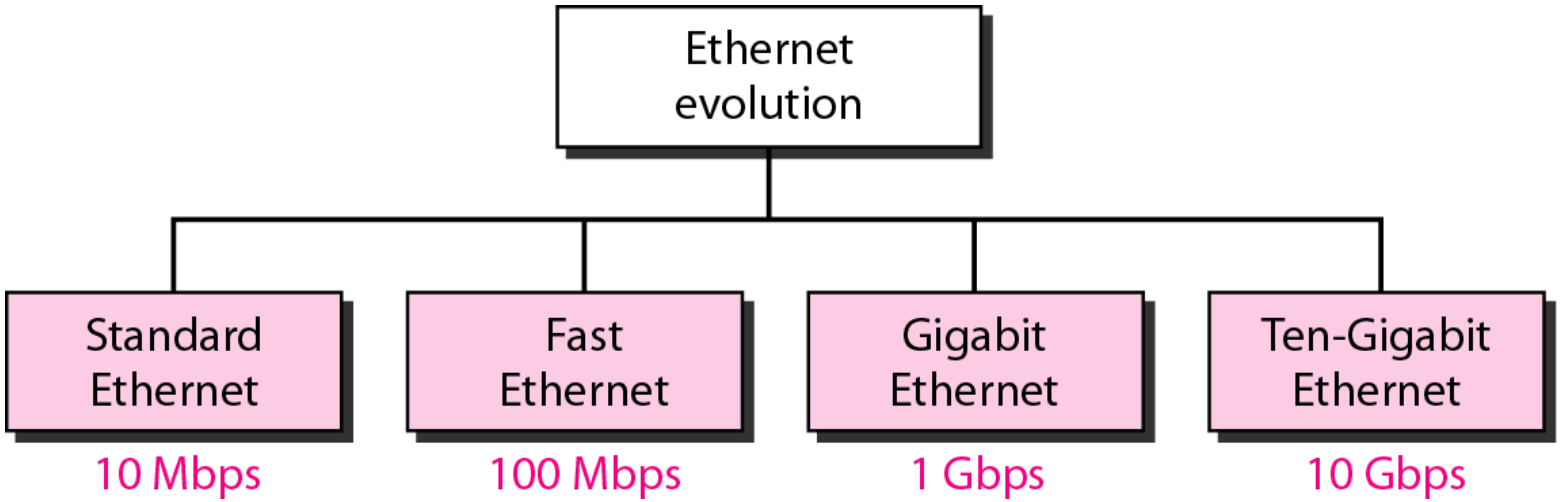


Figure ***Ethernet evolution through four generations***

MAC Sublayer

- 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 the CRC.

- Ethernet does not provide any mechanism for acknowledging received frames, making it what is known as an unreliable medium. Acknowledgments must be implemented at the higher layers. The format of the MAC frame is shown in below figure

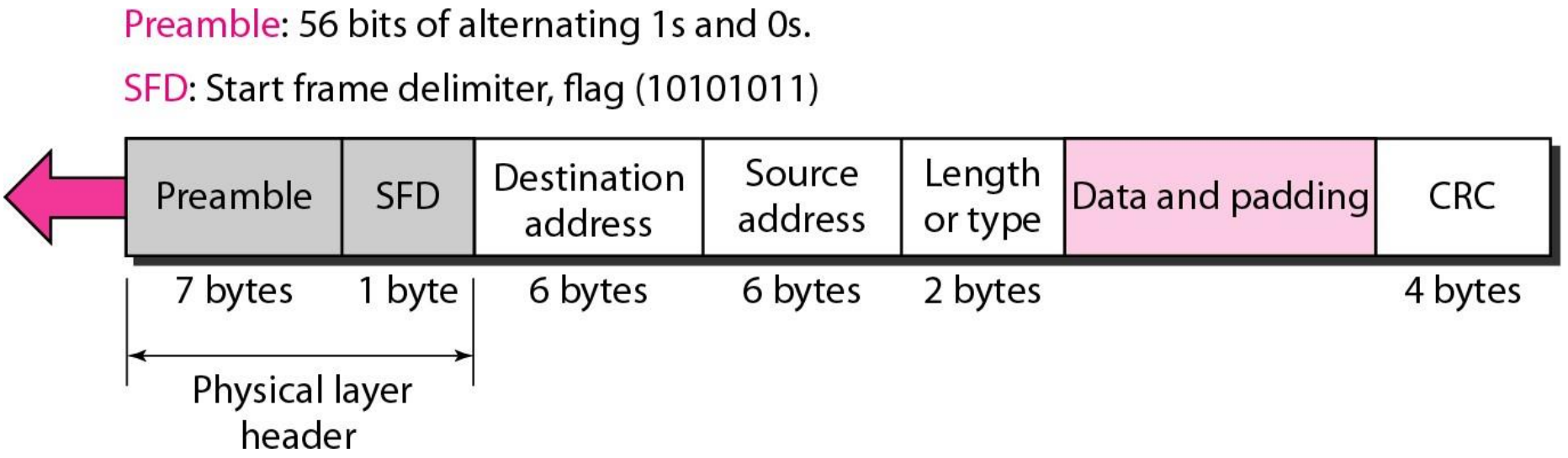


Figure ***802.3 MAC frame***

Preamble. Alerts the receiving system to the coming frame and enables it to synchronize its input timing. 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 station or 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 or 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. The IEEE standard used it as the length field to define the number of bytes in the data field. Both uses are common today.

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

CRC. The last field contains error detection information, in this case a CRC-32

Frame Length

- Ethernet has imposed restrictions on both the minimum and maximum lengths of a frame, as shown in below Figure

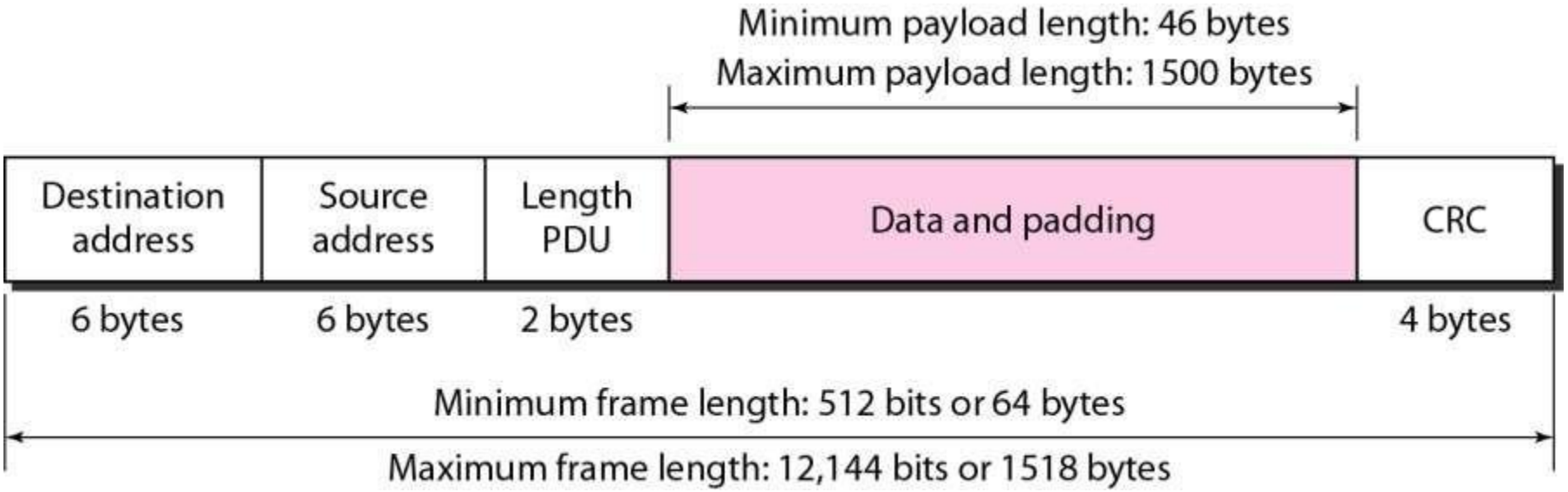


Figure ***Minimum and maximum lengths***

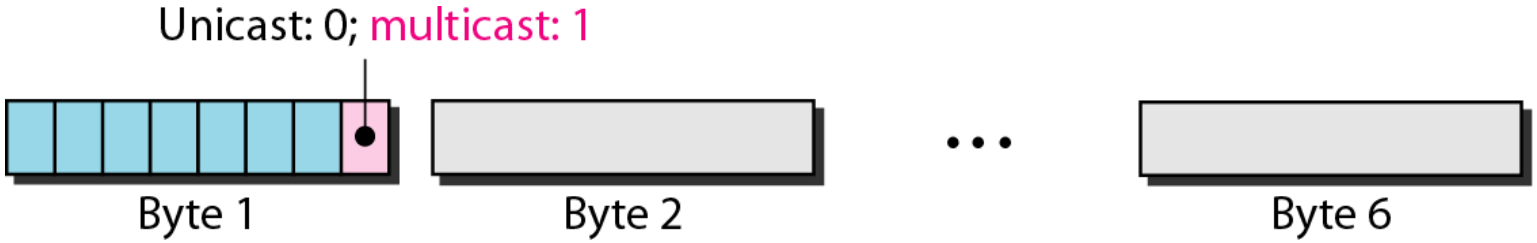
Addressing

- The Ethernet address is 6 bytes (48 bits), normally written in hexadecimal notation, with a colon between the bytes.

Figure ***Example of an Ethernet address in hexadecimal notation***

Unicast, Multicast, and Broadcast Addresses A source address is always a unicast address-the frame comes from only one station.The destination address, however, can be unicast, multicast, or broadcast. Below Figure shows how to distinguish a unicast address from a multicast address.

If the least significant bit of the first byte in a destination address is 0, the address is unicast; otherwise, it is multicast.



Unicast and multicast addresses

Physical Layer

The Standard Ethernet defines several physical layer implementations; four of the most common, are shown in Figure

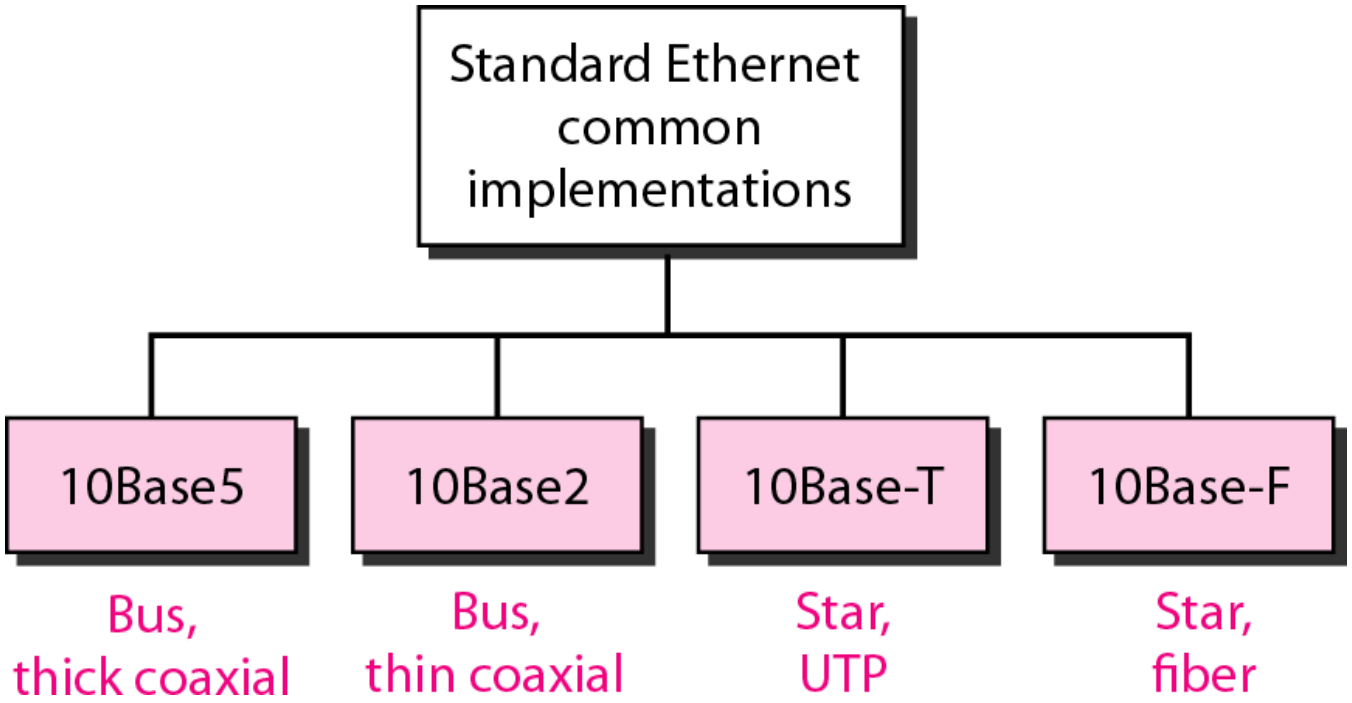


Figure ***Categories of Standard Ethernet***

Encoding and Decoding

- All standard implementations use digital signalling (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. Figure shows the encoding scheme for Standard Ethernet

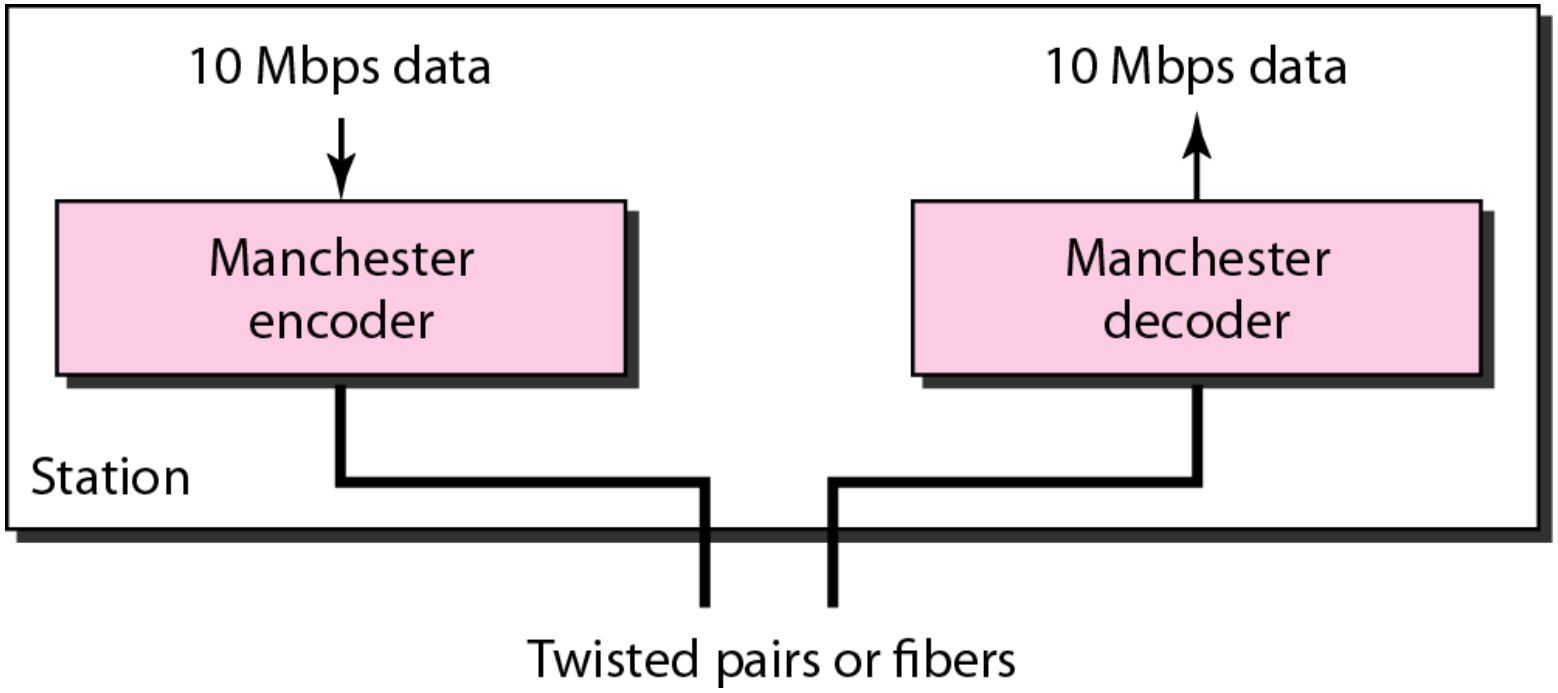


Figure ***Encoding in a Standard Ethernet implementation***

lOBase5: Thick Ethernet

lOBase5 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. Figure shows a schematic diagram of a lOBase5 implementation

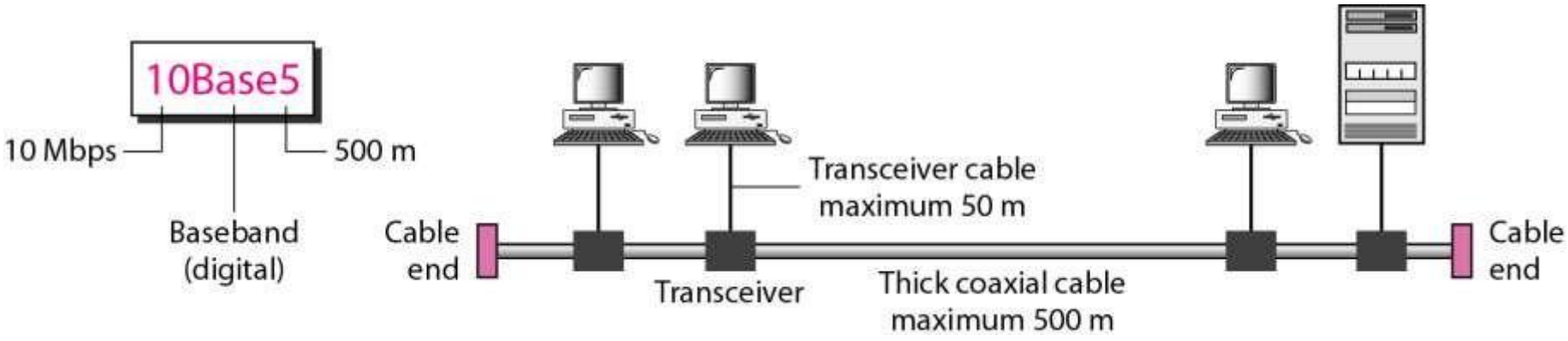


Figure ***10Base5 implementation***

10Base2: Thin Ethernet

The second implementation is called 10 Base2, thin Ethernet, or Cheapernet.10Base2 also uses a bus topology, but the cable is much thinner and more flexible. Figure shows the schematic diagram of a 10Base2 implementation.

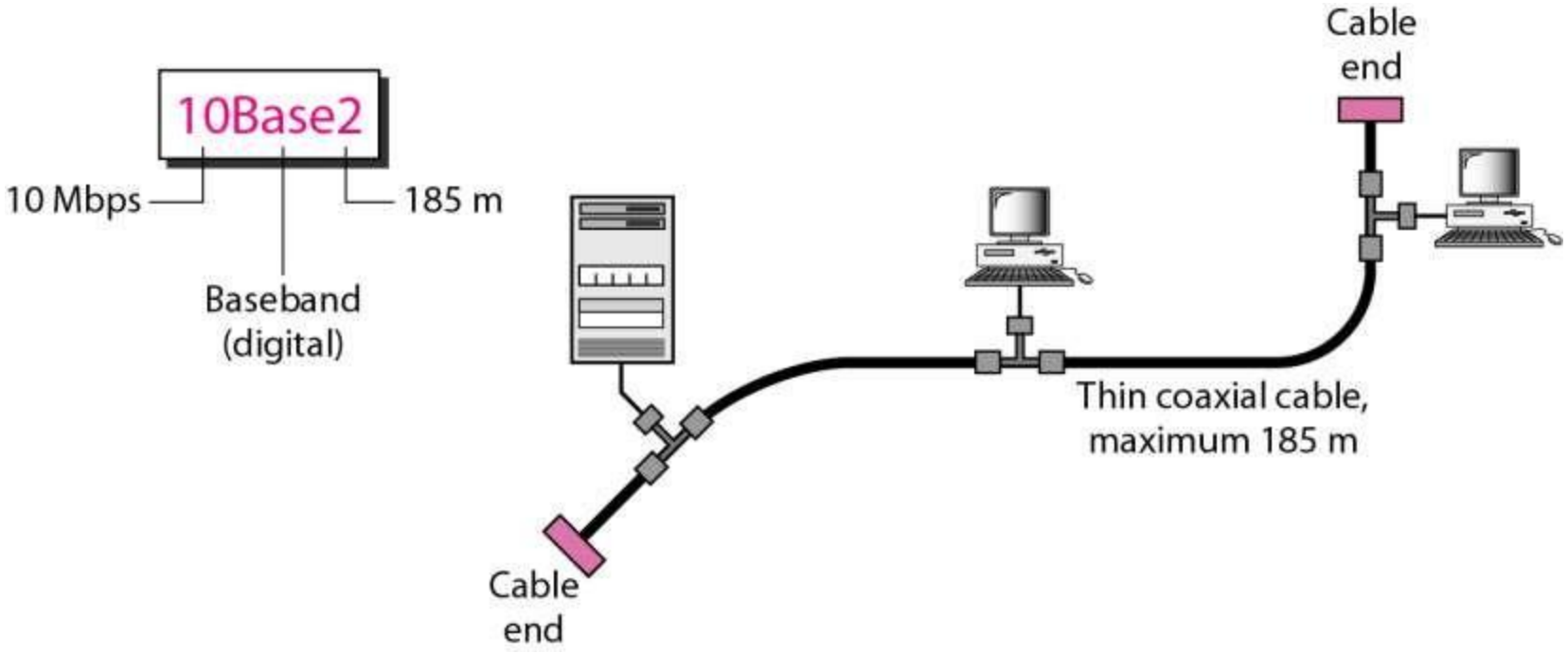


Figure ***10Base2 implementation***

10Base-T: Twisted-Pair Ethernet

- It uses a physical star topology. The stations are connected to a hub via two pairs of twisted cable, as shown in Figure
- The maximum length of the twisted cable here is defined as 100 m, to minimize the effect of attenuation in the twisted cable

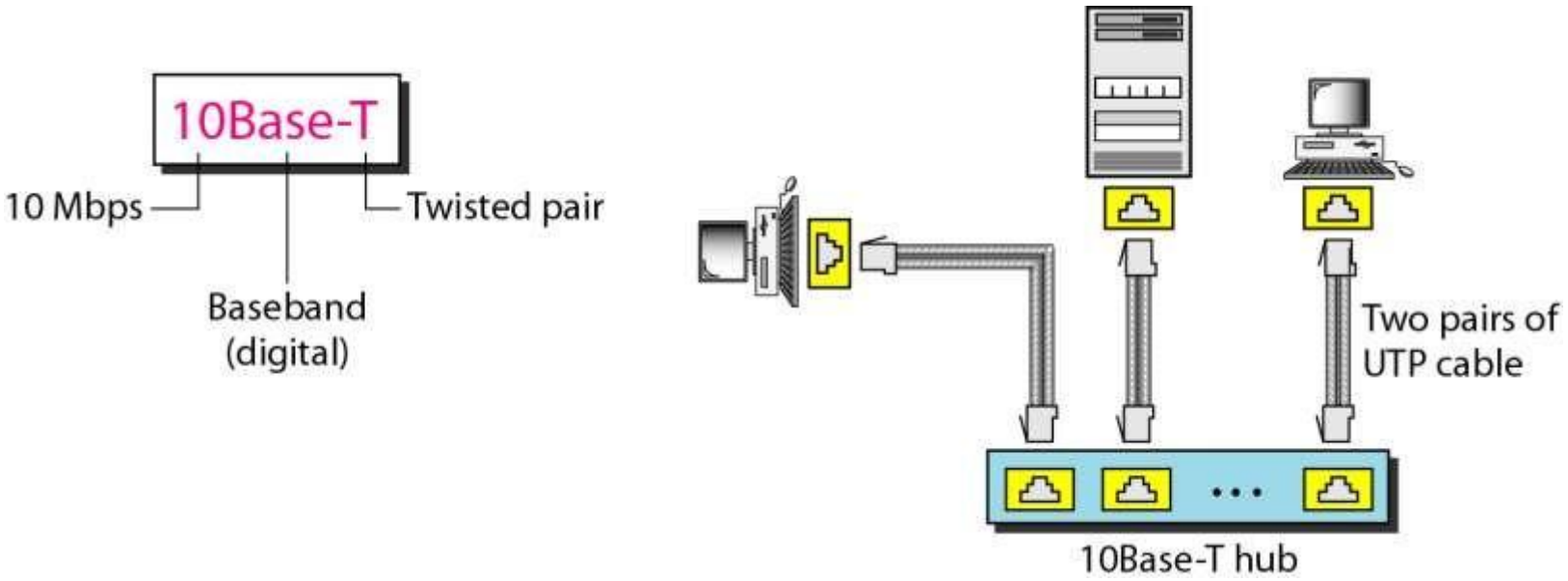


Figure ***10Base-T implementation***

- Although there are several types of optical fiber 10-Mbps Ethernet, the most common is called 10Base-F.
- 10Base-F uses a star topology to connect stations to a hub. The stations are connected to the hub using two fiber-optic cables, as shown in Figure

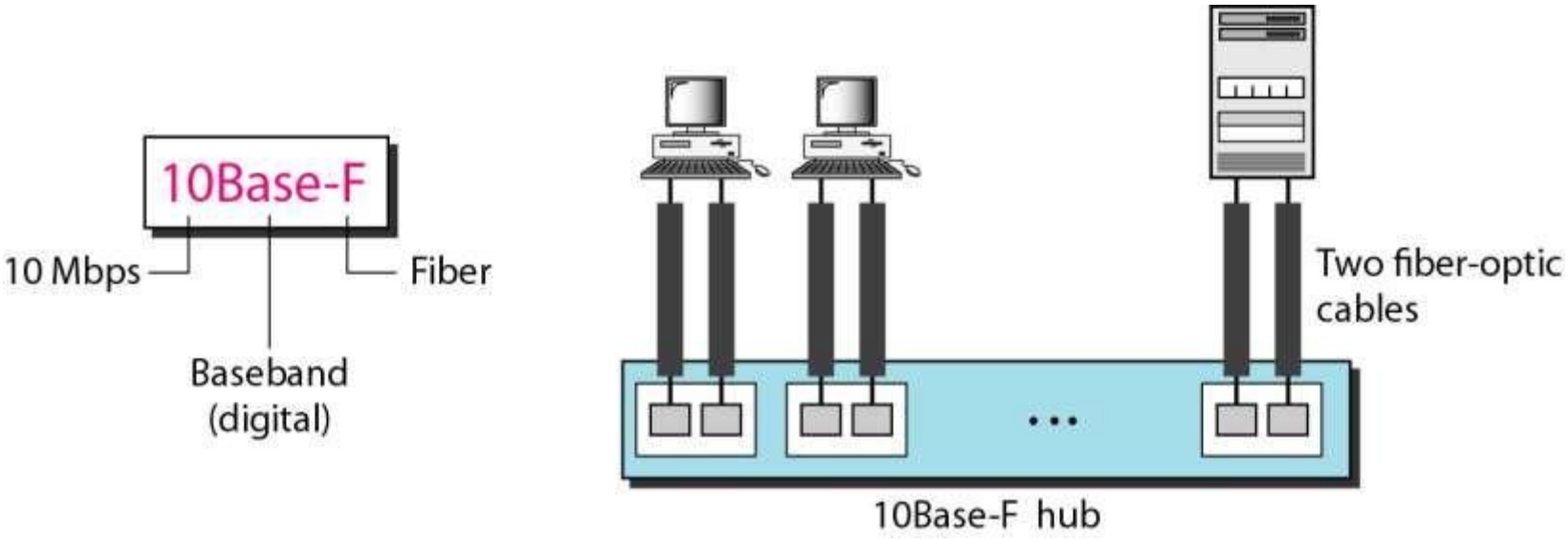
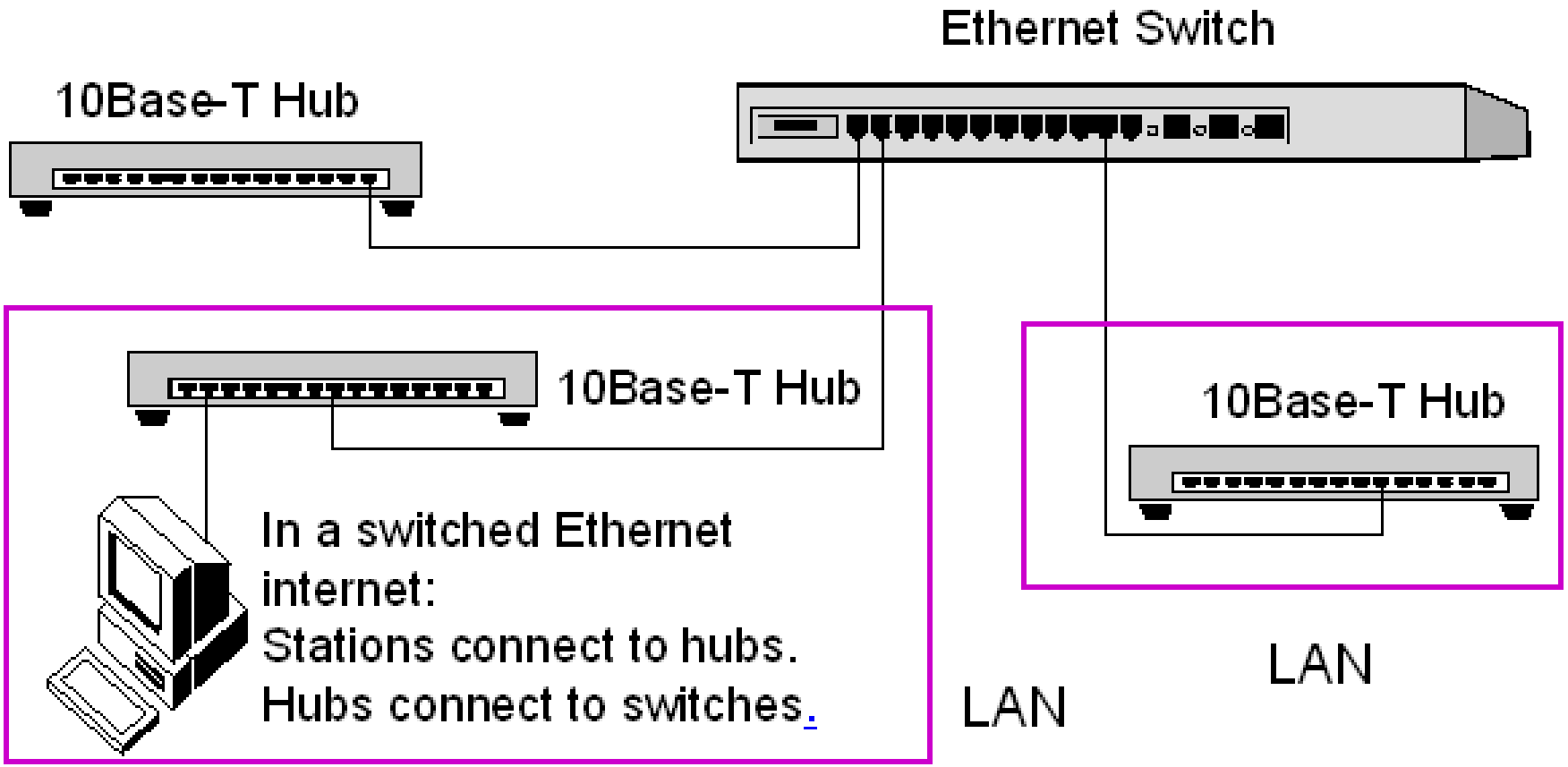


Figure *10Base-F implementation*

Simple Local Internet Using Ethernet Switching and 10Base-T



- Although there are several types of optical fiber 10-Mbps Ethernet, the most common is called 10Base-F.
- 10Base-F uses a star topology to connect stations to a hub. The stations are connected to the hub using two fiber-optic cables, as shown in Figure

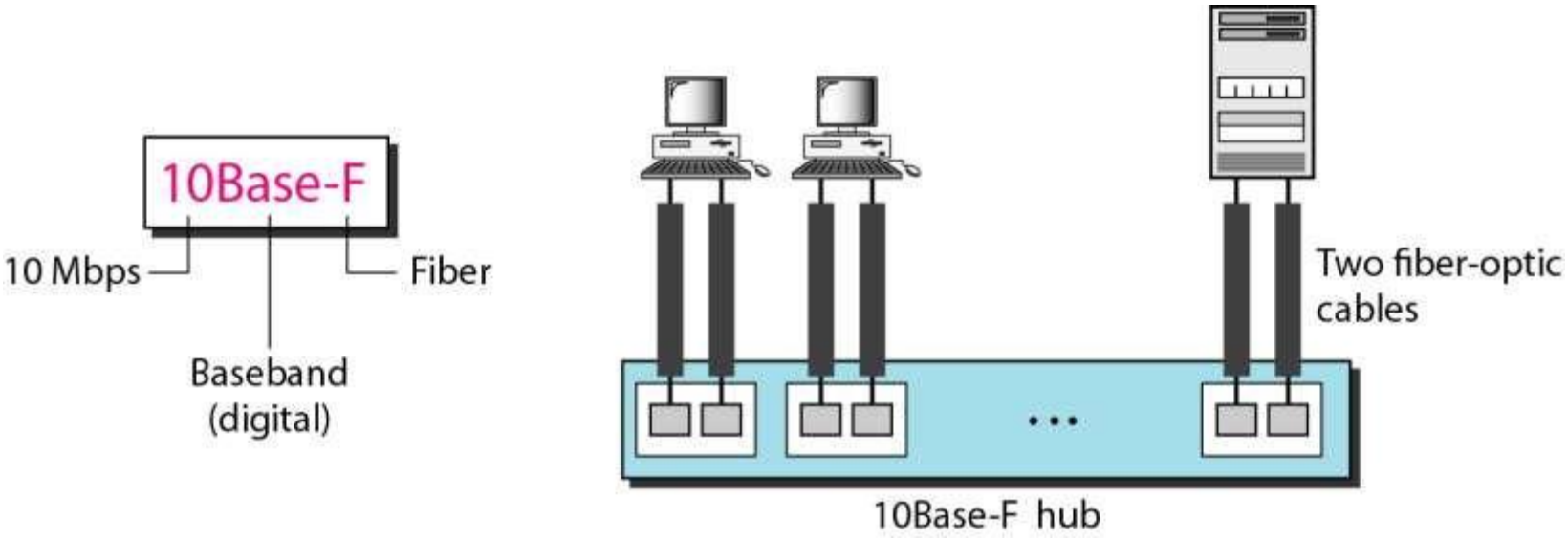


Figure *10Base-F implementation*

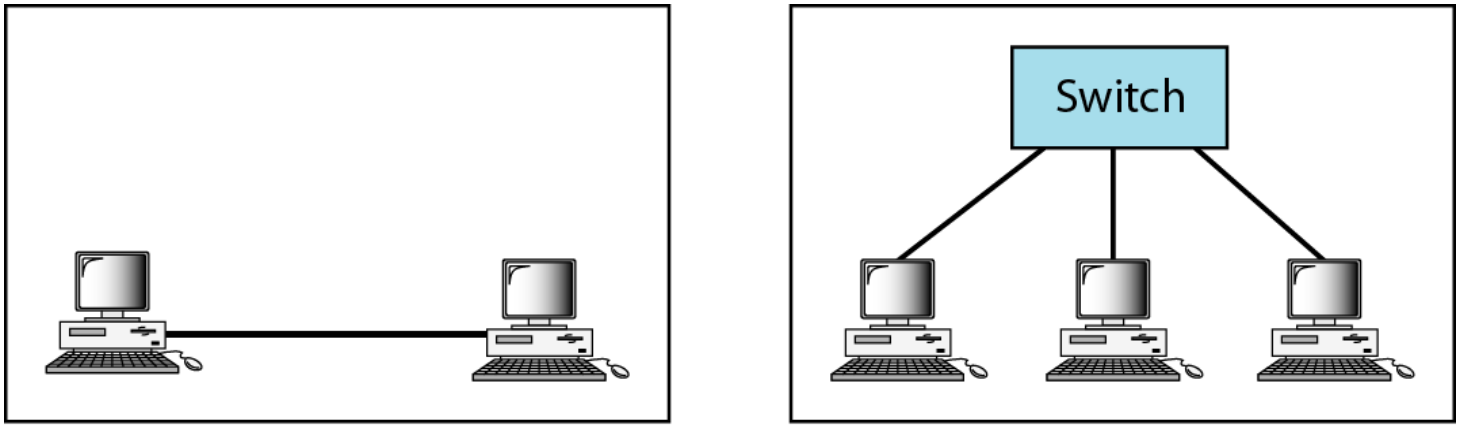
Table 13.1 *Summary of Standard Ethernet implementations*

Characteristics	*10Base5*	*10Base2*	*10Base-T*	*10Base-F*
Media	Thick coaxial cable	Thin coaxial cable	2 UTP	2 Fiber
Maximum length	500 m	185 m	100 m	2000 m
Line encoding	Manchester	Manchester	Manchester	Manchester

FAST ETHERNET

Fast Ethernet was designed to compete with LAN protocols such as FDDI or Fiber Channel. IEEE created Fast Ethernet under the name 802.3u. Fast Ethernet is backward-compatible with Standard Ethernet, but it can transmit data 10 times faster at a rate of 100 Mbps.

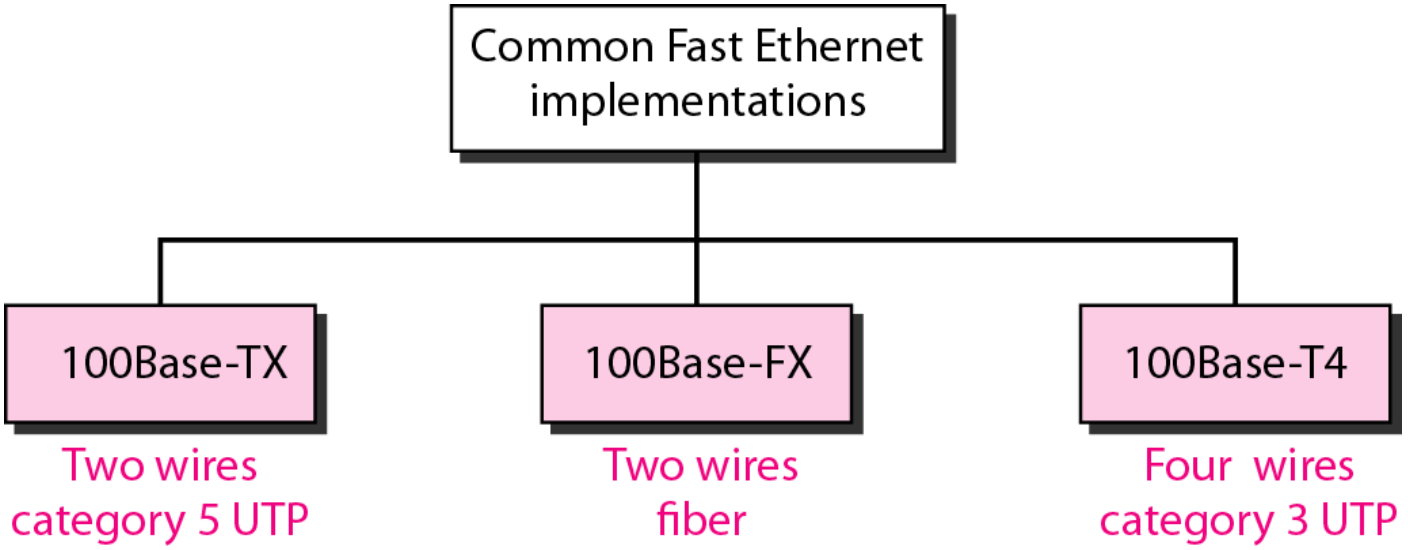
Figure *Fast Ethernet topology*



a. Point-to-point

b. Star

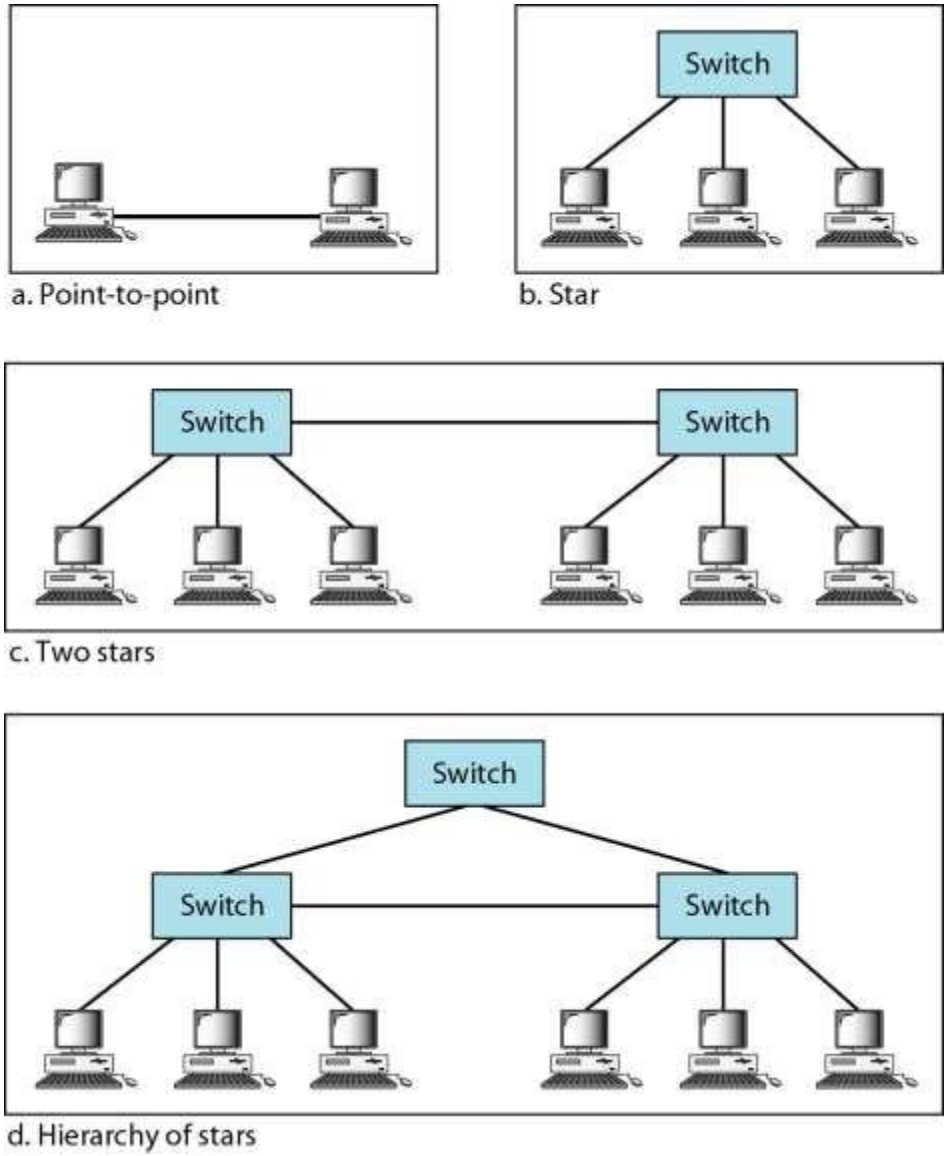
Figure *Fast Ethernet implementations*



GIGABIT ETHERNET

The need for an even higher data rate resulted in the design of the Gigabit Ethernet protocol (1000 Mbps). The IEEE committee calls the standard 802.3z.

Figure *Topologies of Gigabit Ethernet*



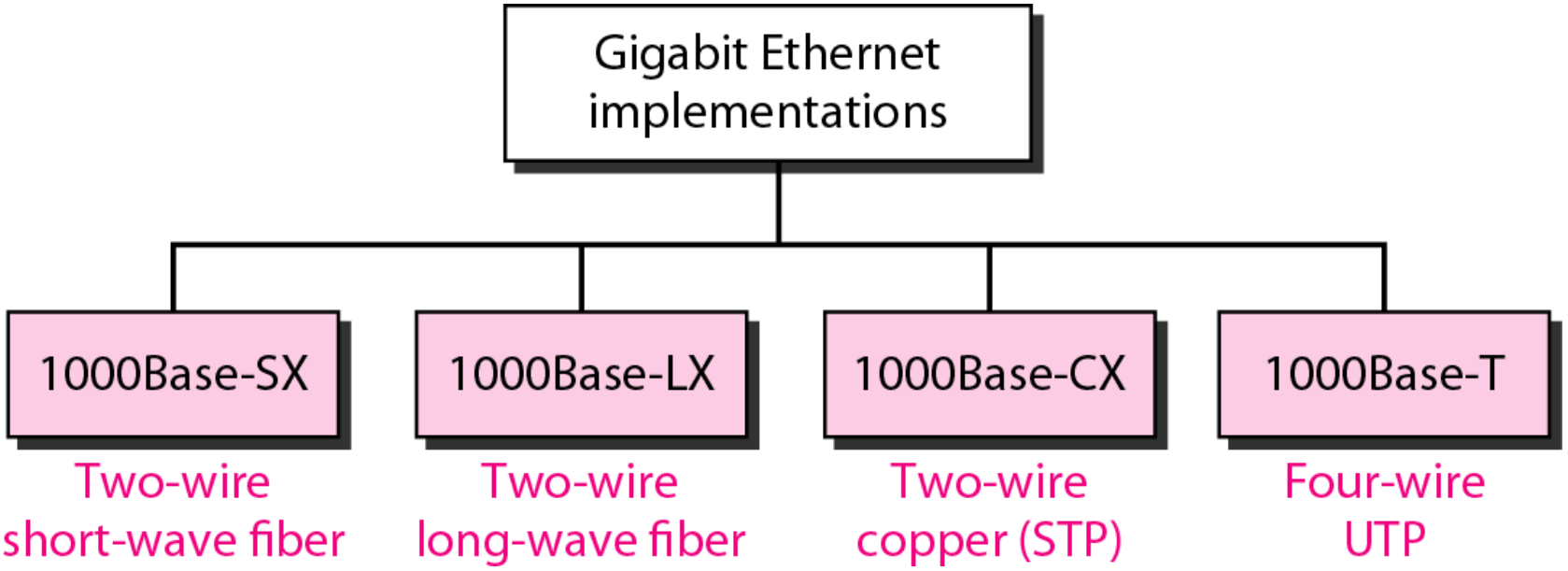
a. Point-to-point

b. Star

c. Two stars

d. Hierarchy of stars

Figure *Gigabit Ethernet implementations*



Summary of Gigabit Ethernet implementations

Characteristics	*1000Base-SX*	*1000Base-LX*	*1000Base-CX*	*1000Base-T*
Media	Fiber short-wave	Fiber long-wave	STP	Cat 5 UTP
Number of wires	2	2	2	4
Maximum length	550 m	5000 m	25 m	100 m
Block encoding	8B/10B	8B/10B	8B/10B	
Line encoding	NRZ	NRZ	NRZ	4D-PAM5

Summary of Ten-Gigabit Ethernet implementations

Characteristics	*10GBase-S*	*10GBase-L*	*10GBase-E*
Media	Short-wave 850-nm multimode	Long-wave 1310-nm single mode	Extended 1550-mm single mode
Maximum length	300 m	10 km	40 km

Datalink Layer Switching

- Network switching is the process of forwarding data frames or packets from one port to another leading to data transmission from source to destination.

- Data link layer is the second layer of the Open System Interconnections (OSI) model whose function is to divide the stream of bits from physical layer into data frames and transmit the frames according to switching requirements.

- Switching in data link layer is done by network devices called **bridges**.

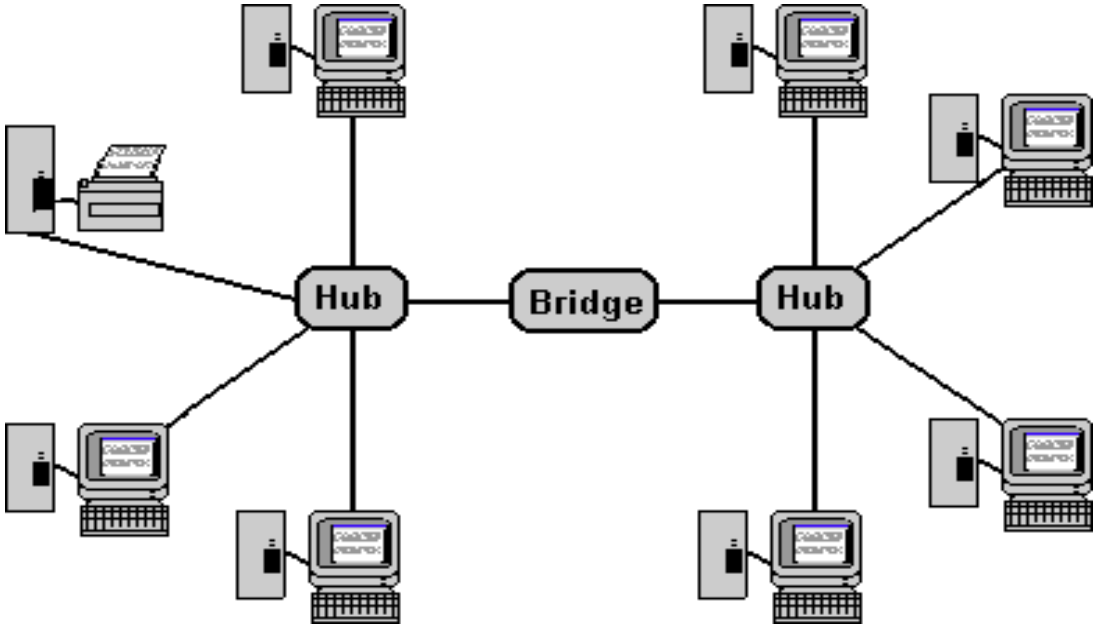
Datalink Layer Switching

- Uses of bridges
- Learning bridges
- Spanning Tree bridges
-Repeaters, hubs, bridges, switches, routers, and gateways

Uses of bridges

- A bridge is a network device that connects multiple LANs (local area networks) together to form a larger LAN.

- The process of aggregating networks is called network bridging. A bridge connects the different components so that they appear as parts of a single network.



- By joining multiple LANs, bridges help in multiplying the network capacity of a single LAN.

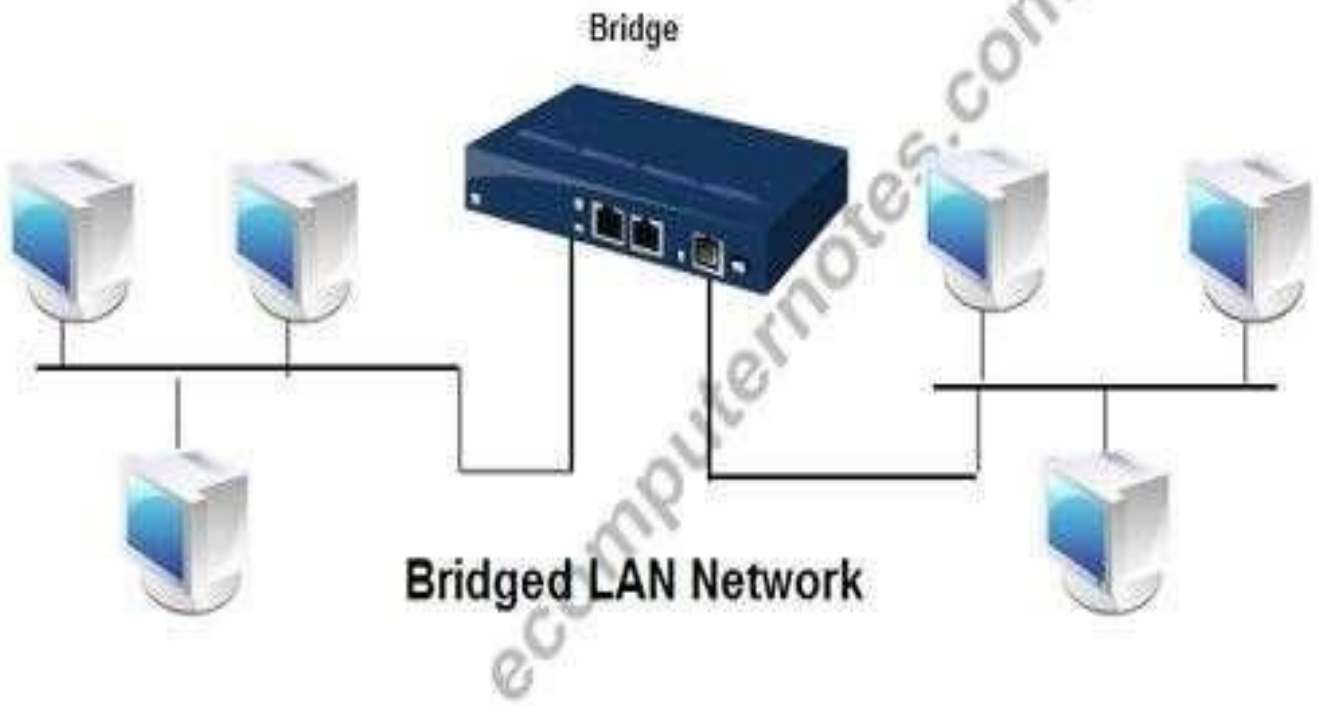
- Since they operate at data link layer, they transmit data as data frames. On receiving a data frame, the bridge consults a database to decide whether to pass, transmit or discard the frame.

  - ✓ If the frame has a destination MAC (media access control) address in the same network, the bridge passes the frame to that node and then discards it.

  - ✓ If the frame has a destination MAC address in a connected network, it will forward the frame toward it.

Learning Bridges

- Bridge is a device that joins networks to create a much larger network. ... A learning bridge, also called an adaptive bridge, "learns" which network addresses are on one side of the bridge and which are on the other so it knows how to forward packets it receives.



The Learning Algorithm

The Learning Algorithm can be written in Pseudocode as follows:

If the address is in the tables then
Forward the packet onto the necessary port.

If the address is not in the tables, then
Forward the packet onto every port except for the port that the packet was received on, just to make sure the destination gets the message.

Add an entry in your internal tables linking the Source Address of the packet to whatever port the packet was received from.

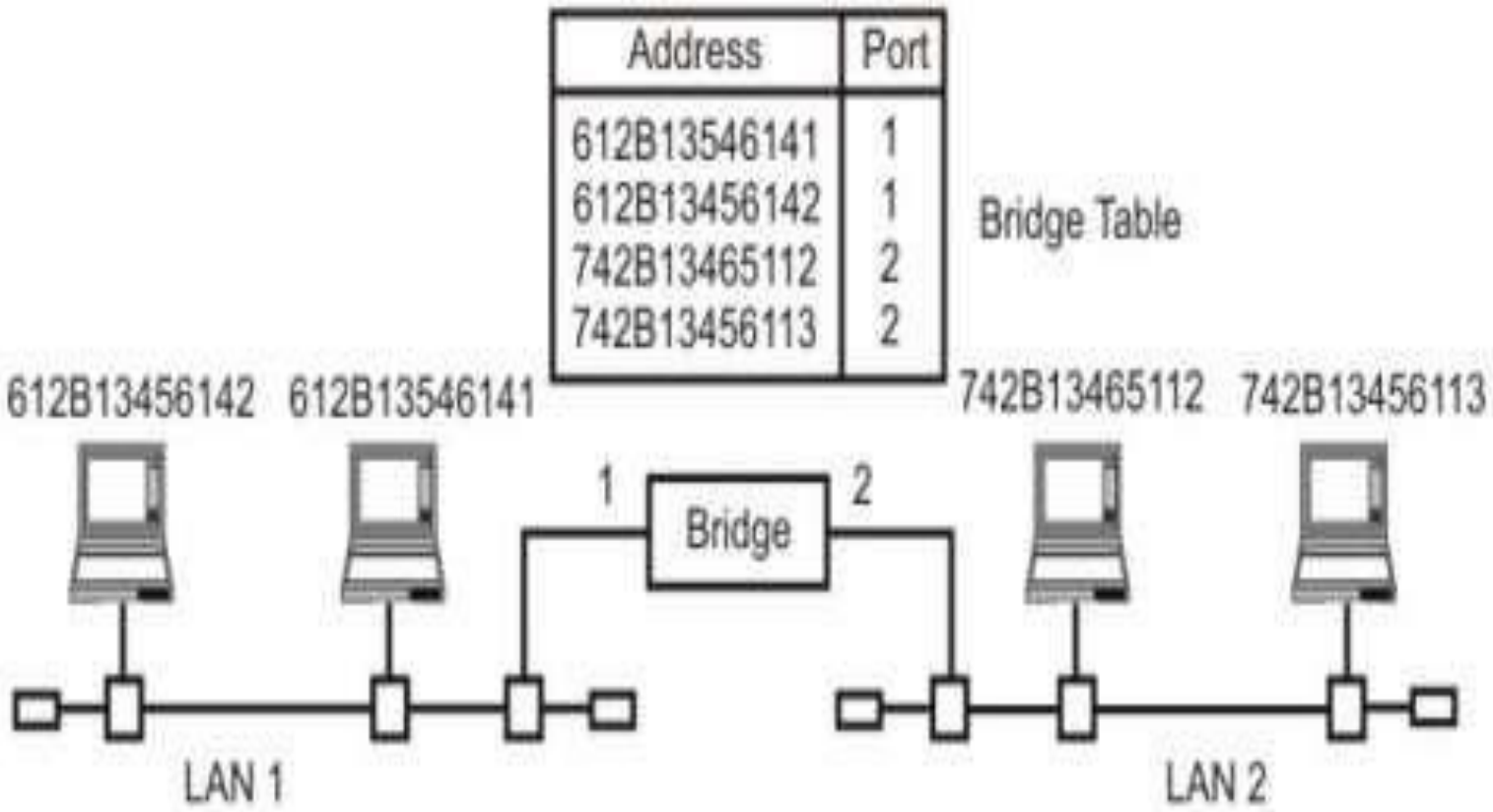


Figure 1.3 A bridge connecting two separate LANs

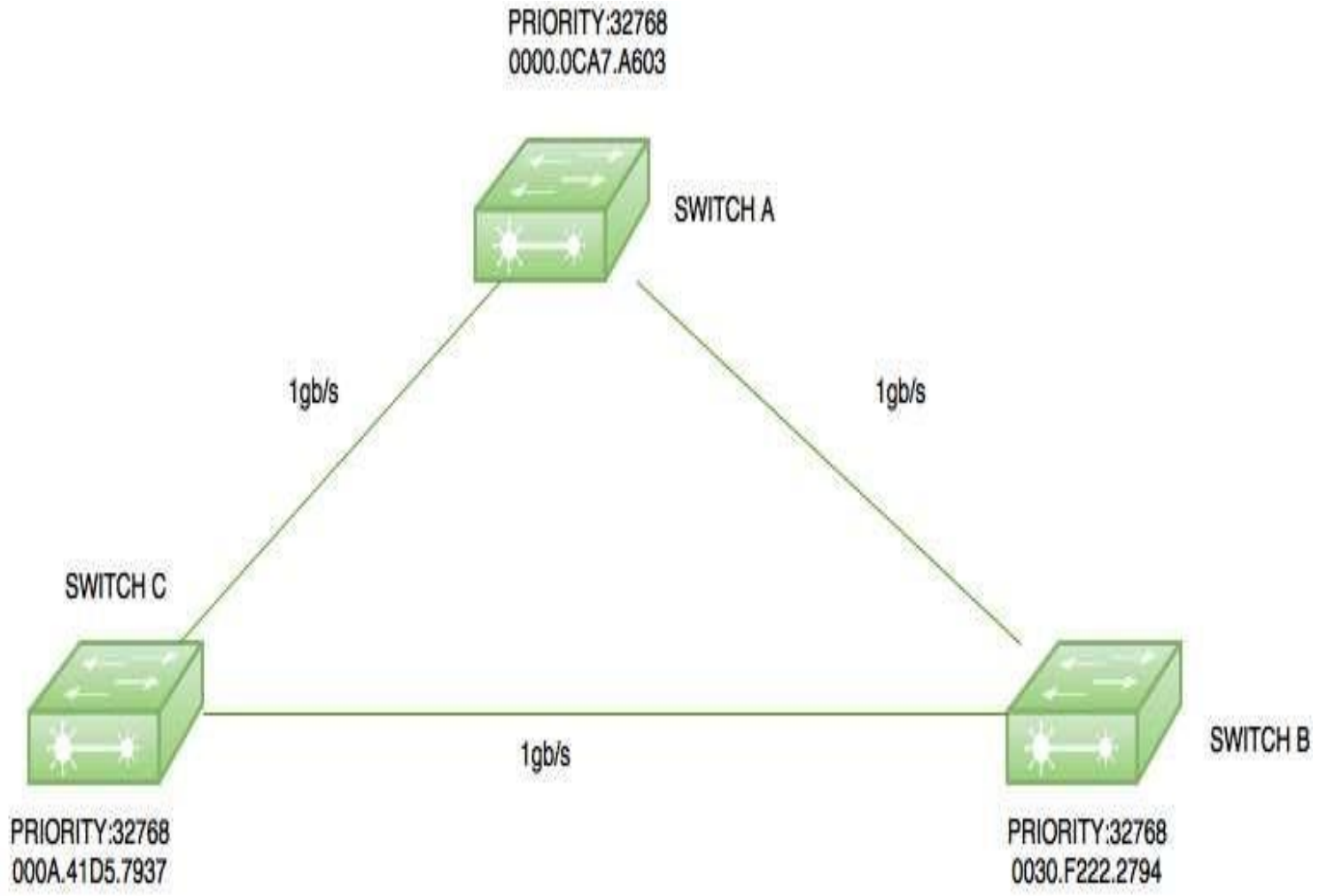
Spanning Tree Bridges

- Redundant links are used to provide back up path when one link goes down but Redundant link can sometime cause switching loops.

- The main purpose of Spanning Tree Protocol (STP) is to ensure that you do not create loops when you have redundant paths in your network.

- The Spanning Tree Protocol (STP) is a network protocol that builds a loop-free logical topology for Ethernet networks. Means it was created to prevent loops

Lets understand what is loop ?



Disadvantage of the loop

- This process can saturate all bandwidth of the network by creating and forwarding the multiple copies of the same frame. It also significantly decreases the performance of the end devices by forcing them to process duplicate copies of the same frame.

Then how to identify loop?

- Switches send probes into the network to discover loops. These probes are called as BPDU(Bridge Priority Data Unit)

- Switch multicasts BPDU probes and if it receives its own BPDU back , it means there is a loop in the network

How to avoid loops?

All switches will find the best way to **elect a root bridge** and the redundant links will be blocked

Bridge Priority Data Unit (BPDU) – It contains the Sender's Bridge I'd, Cost to the Root Bridge. The switch with the lowest Bridge I'd will become the root bridge.

Bridge I'd – It is a 8-byte field which is a combination of bridge priority (2 bytes) and Base Mac address (6 bytes) of a device. If there is a tie on bridge priority then the Base Mac address is considered.

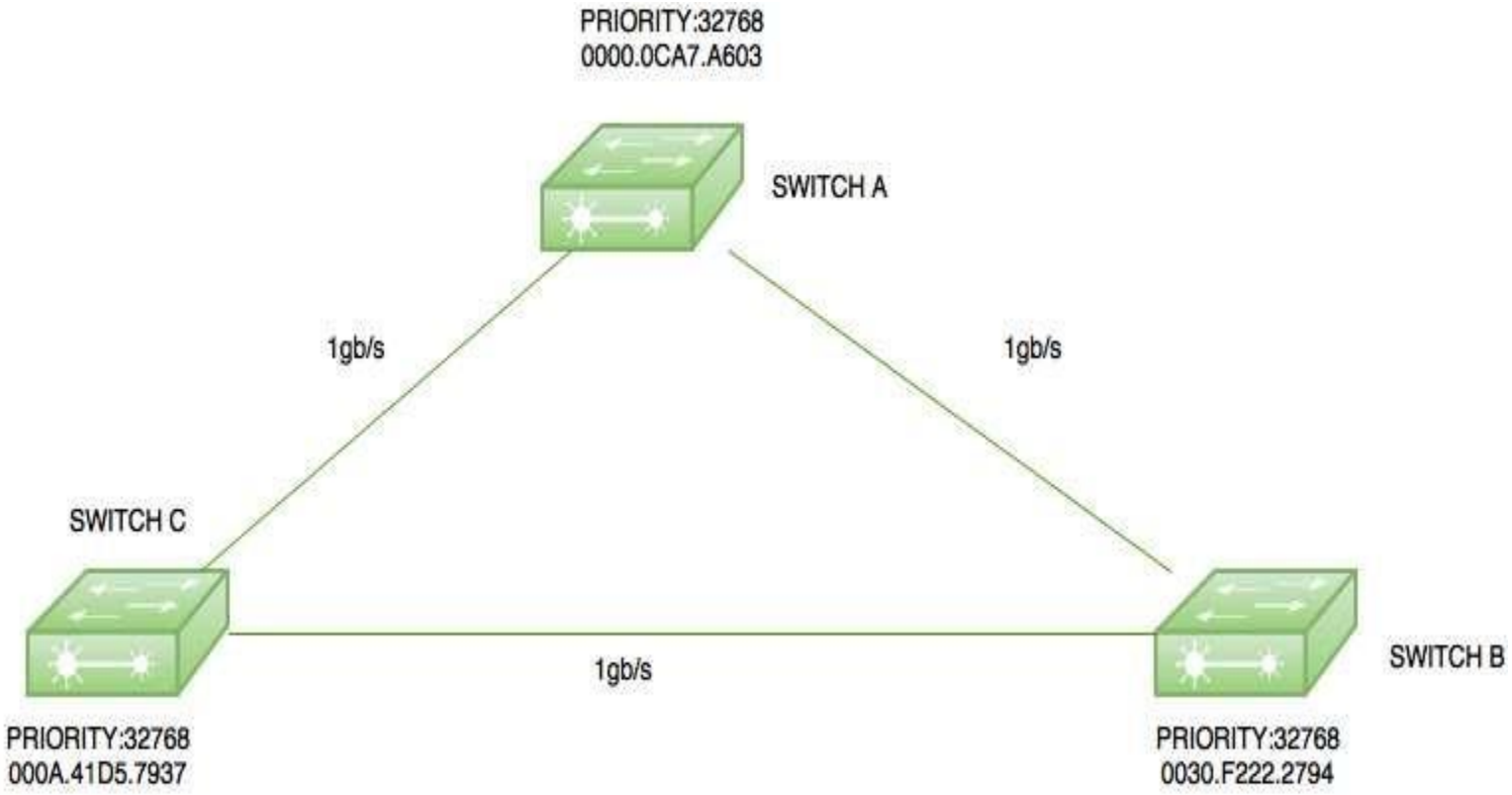
Bridge Priority – It is priority, which is assigned to every switch, 32768 by default.

Root Bridge – The root bridge is the bridge with lowest Bridge I'd. All the decisions like which ports are the root ports (the port with best path to the root bridge) are made from the perspective of root bridge.

Path cost – A switch may encounter one or more switch in the path to the root bridge. All the paths are analysed and the path with the lowest cost will be selected.

SPEED	LINK COST
10 Mbps	100
100 Mbps	19
1 Gbps	4
10 Gbps	2

Example -



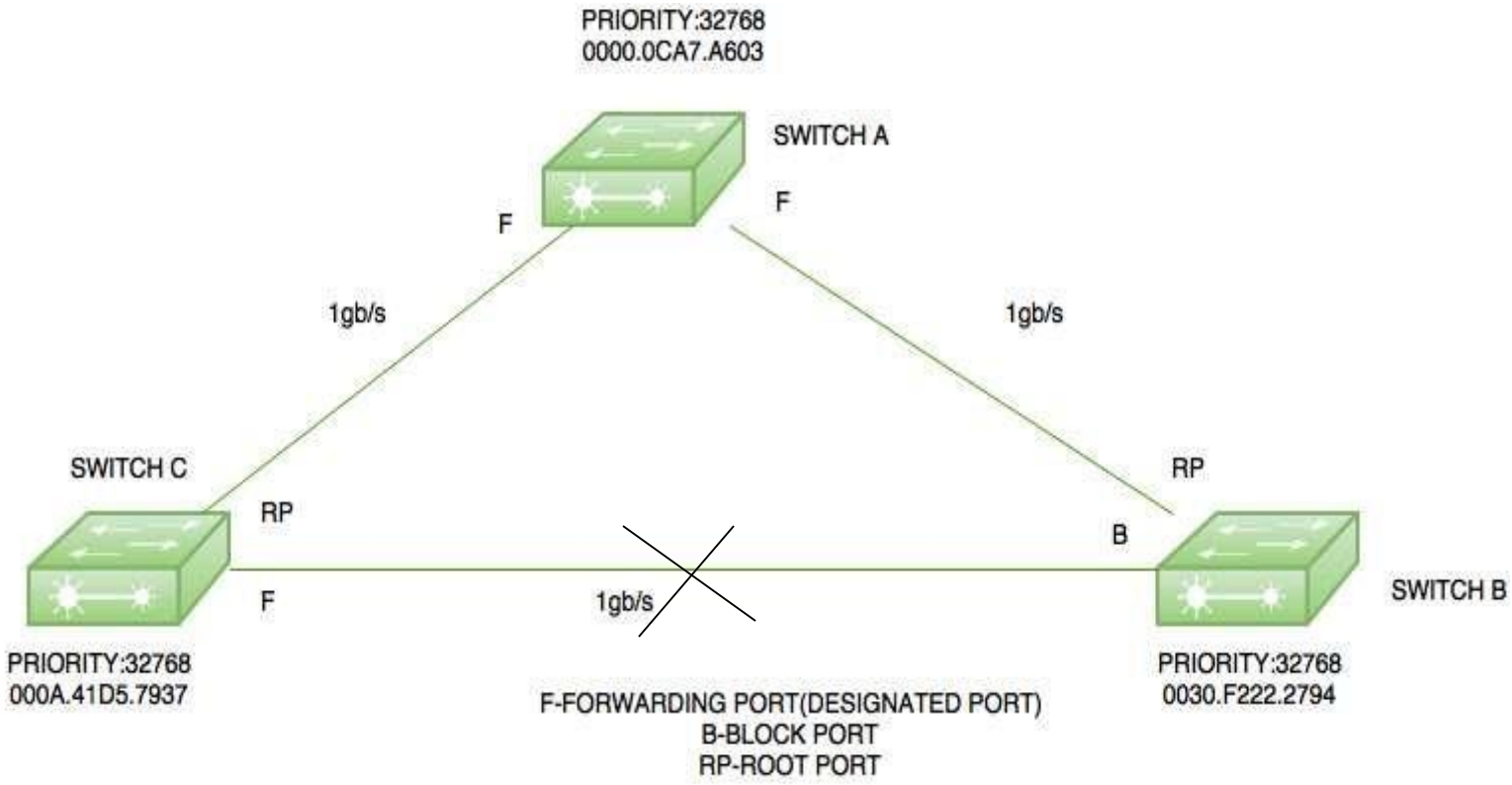
Here is a small topology with three switches switch A (mac address-0000.0ACA7.A603), switch B(0030.F222.2794) and switch C(000A.41D5.7937) with all having default priority (32768).

Root Bridge election –

As all the switches have default priority therefore there is a tie on the basis of priority. Now, the switch with the lowest Mac address will become a root bridge. Here, switch A will become the root bridge as it has the lowest Mac address. Therefore, the ports of switch A will be in forwarding state i.e designated port.

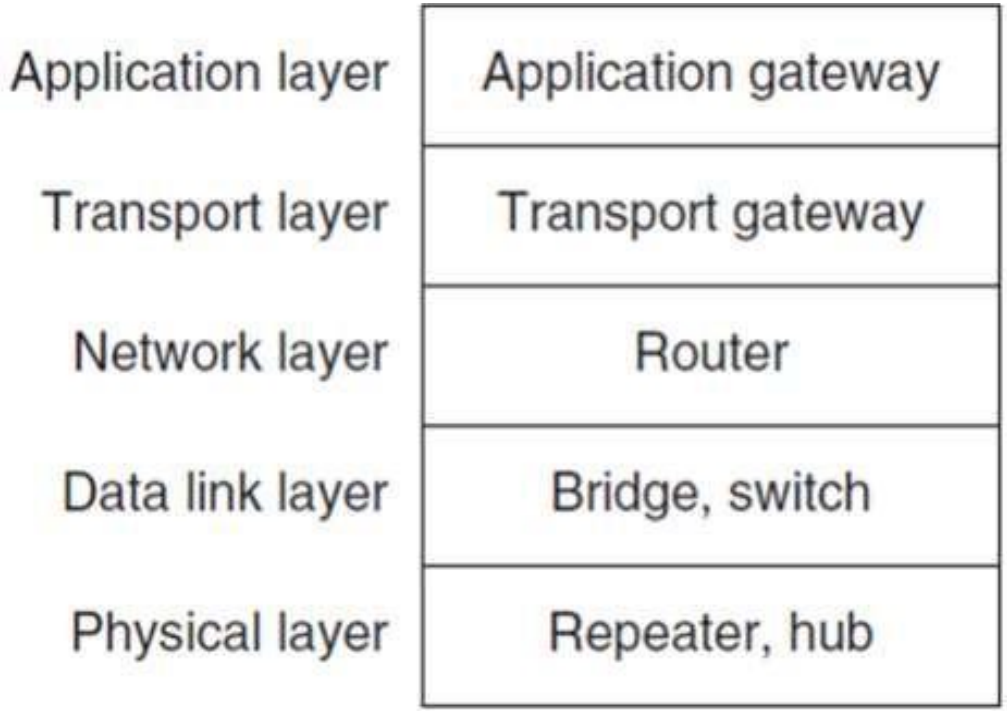
Root Ports Election –

If switch C choose the path through switch B then the cost will be (4+4=8) but if it chooses the directly connected path to switch A then the cost will be 4 therefore both switch B and switch C will choose the ports connected to switch A as their root ports.



Now as the link between switch B and switch C as the same cost to the root bridge therefore the switch with lowest bridge I'd will be in forwarding mode therefore switch C port will be in forwarding mode and switch B port will be in block mode.

Repeaters, Hubs, Bridges, Switches, Routers, and Gateways



Hub

Gateway

Router

Repeater

Bridge

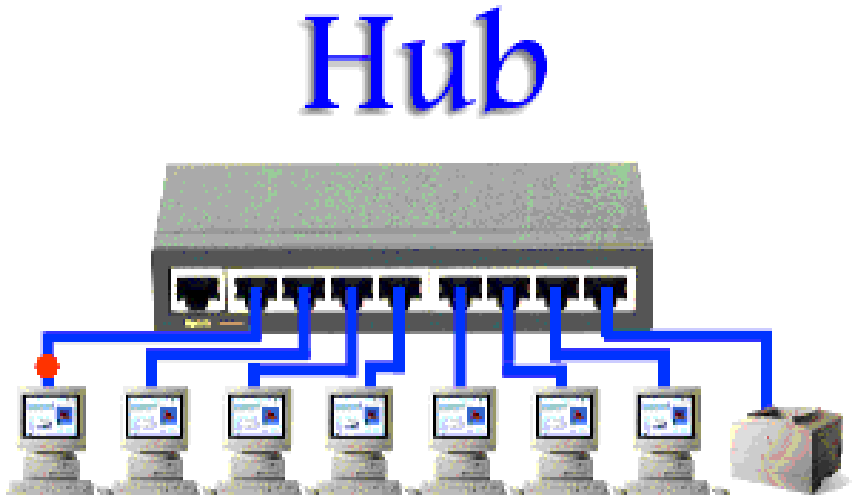
Switch

1. Repeater

- A repeater operates at the physical layer. Its job is to regenerate the signal over the same network before the signal becomes too weak or corrupted.
- An important point to be noted about repeaters is that they do not amplify the signal. When the signal becomes weak, they copy the signal bit by bit and regenerate it at the original strength. It is a 2 port device.

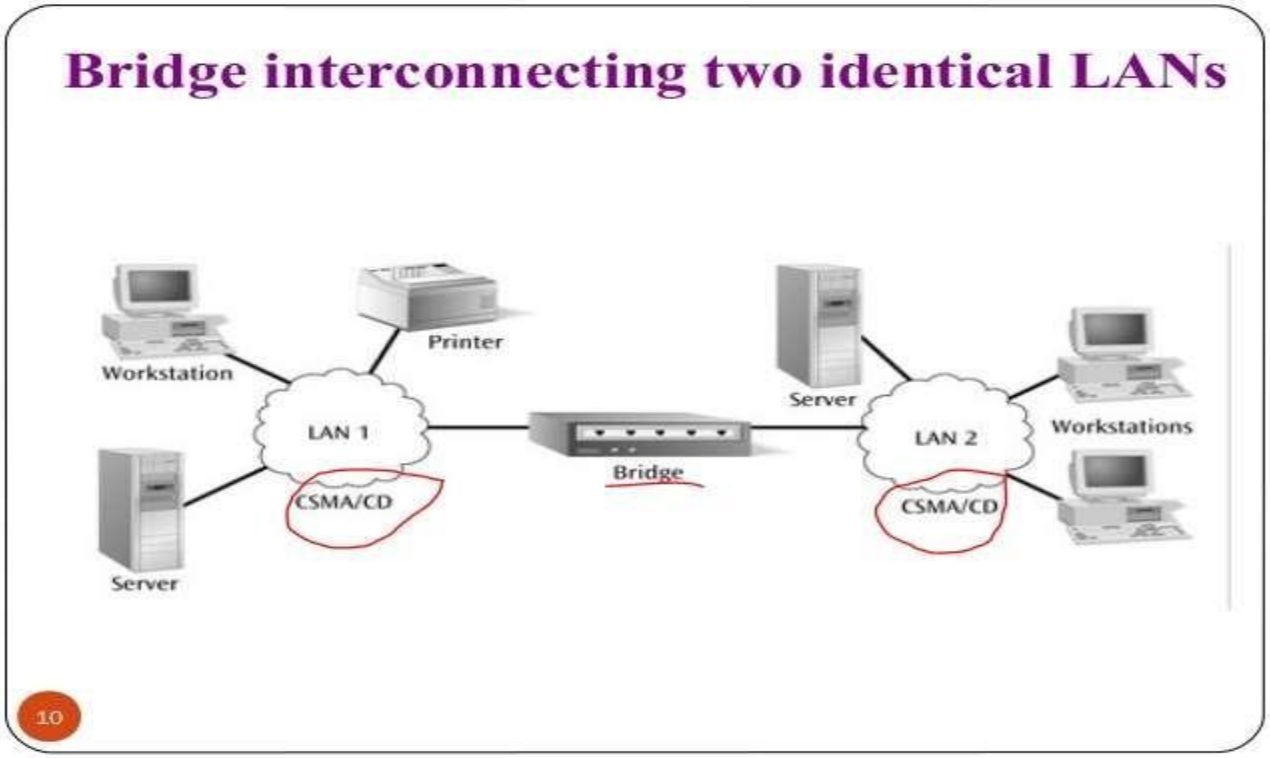
2. Hub

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



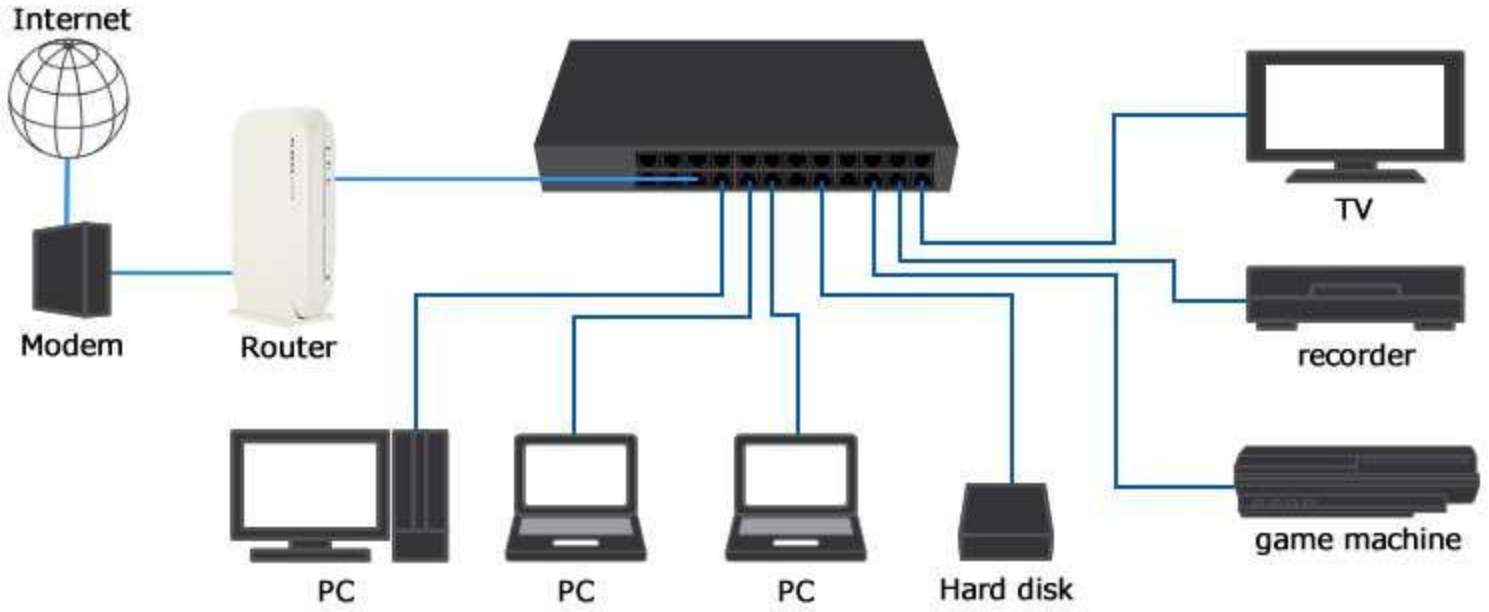
3. Bridge

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

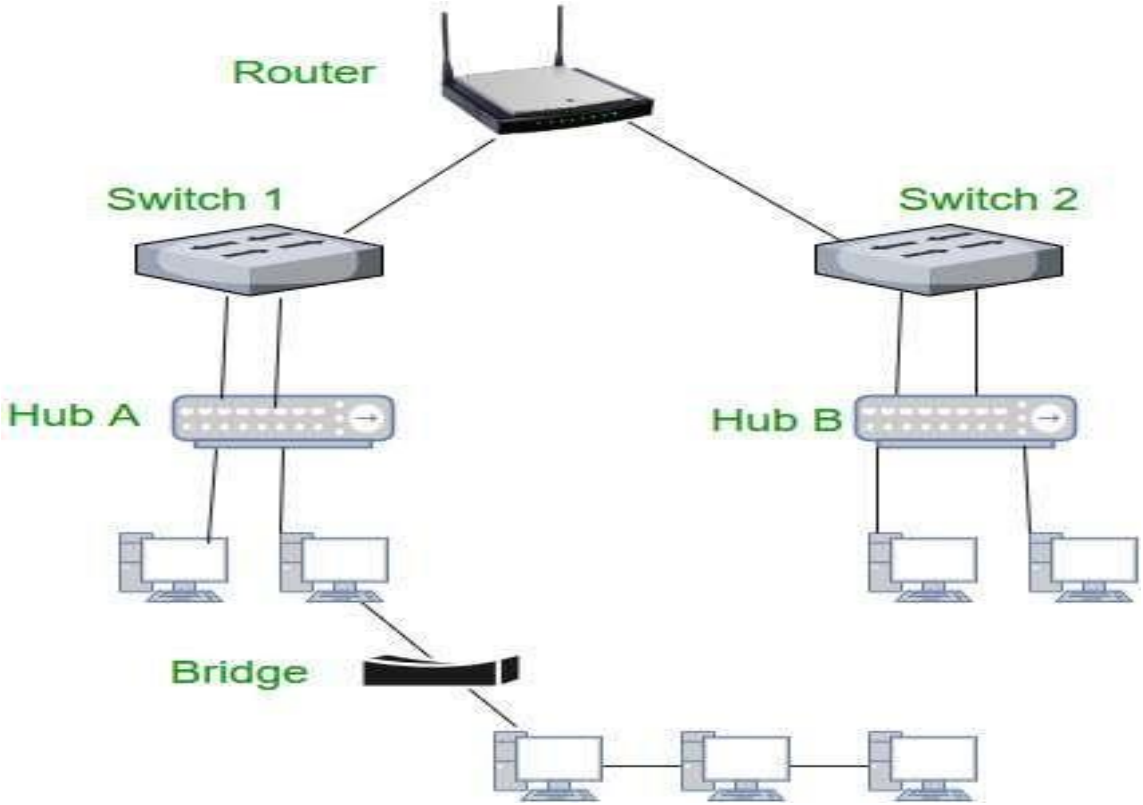


4. Switch

- A **switch** is a device that connects other devices together. Multiple data cables are plugged into a **switch** to enable communication between different networked devices. A switch is a data link layer device.
- The switch can perform error checking before forwarding data, that makes it very efficient as it does not forward packets that have errors and forward good packets selectively to correct port only.



5.Routers A router is a device like a switch that routes data packets based on their IP addresses. Router is mainly a Network Layer device. Routers normally connect LANs and WANs together and have a dynamically updating routing table based on which they make decisions on routing the data packets.



A **gateway** is a hardware **device** that acts as a "gate" between two networks. It may be a router, firewall, server, or other **device** that enables traffic to flow in and out of the network.