Module 4

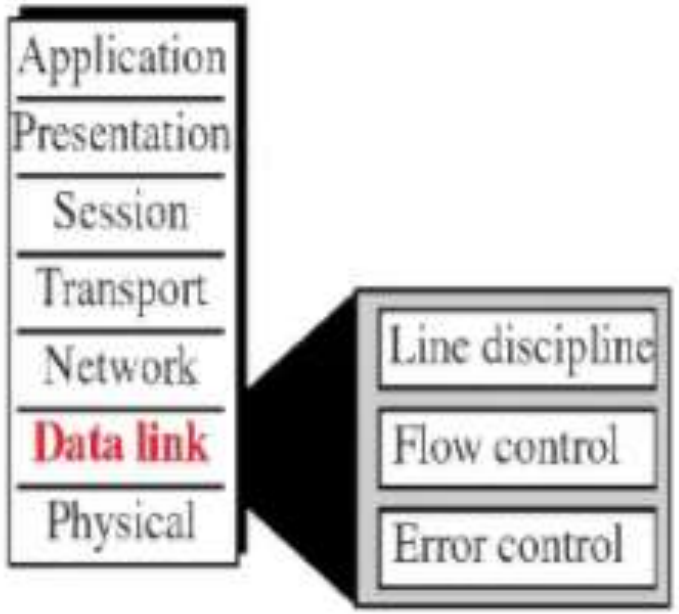
DATA LINK LAYER

CONTENTS

- Introduction
- Framing
- Error detection methods-Parity check,CRC,Checksum

INTRODUCTION

- deals with procedures for communication between two adjacent nodes—node-to-node communication.
- Data link control (DLC) functions include
- Framing
- flow and error Control
- error detection and correction.



FRAMING

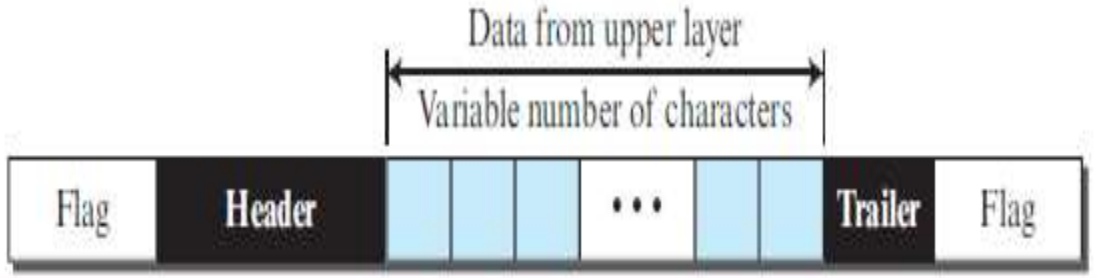
- Data transmission in the physical layer means moving bits in the form of a signal from the source to the destination.

- The data-link layer, on the other hand, needs to pack bits into frames, so that each frame is distinguishable from another

- Framing in the data-link layer separates a message from one source to a destination by adding a sender address and a destination address.

Character-Oriented Framing

Figure 5.4 *A frame in a character-oriented protocol*

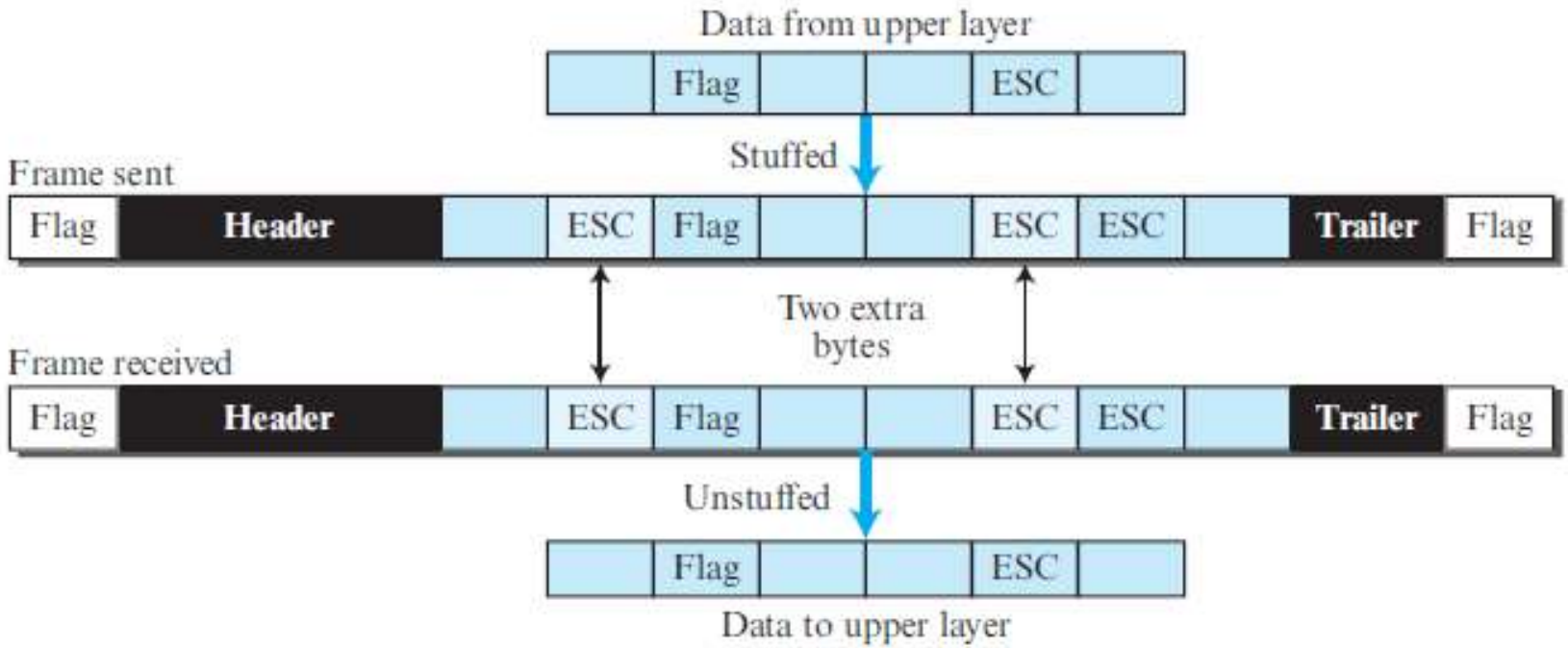


- In character-oriented (or byte-oriented) framing, data to be carried are 8-bit characters
- The header which carries the source and destination addresses and other control information, and the trailer, which carries error detection redundant bits, are also multiples of 8 bits.
- To separate one frame from the next, an 8-bit (1-byte) flag is added at the beginning and the
- end of a frame.
- The flag, composed of protocol-dependent special characters, signals the start or end of a frame.
- Character-oriented framing was used when only text was exchanged by the data-link layers.

Byte stuffing

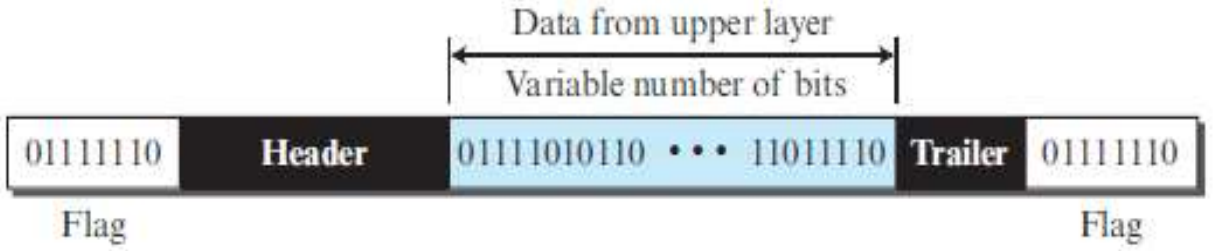
- *Byte stuffing is the process of adding one extra byte whenever there is a flag or escape character in the text*
- In byte stuffing (or character stuffing), a special byte is added to the data section of the frame when there is a character with the same pattern as the flag.
- The data section is stuffed with an extra byte.
- This byte is usually called the escape character (ESC) and has a predefined bit pattern.
- Whenever the receiver encounters the ESC character, it removes it from the data section and treats the next character as data, not as a delimiting flag.

Figure 5.5 *Byte stuffing and unstuffing*



Bit-Oriented Framing

Figure 5.6 *A frame in a bit-oriented protocol*

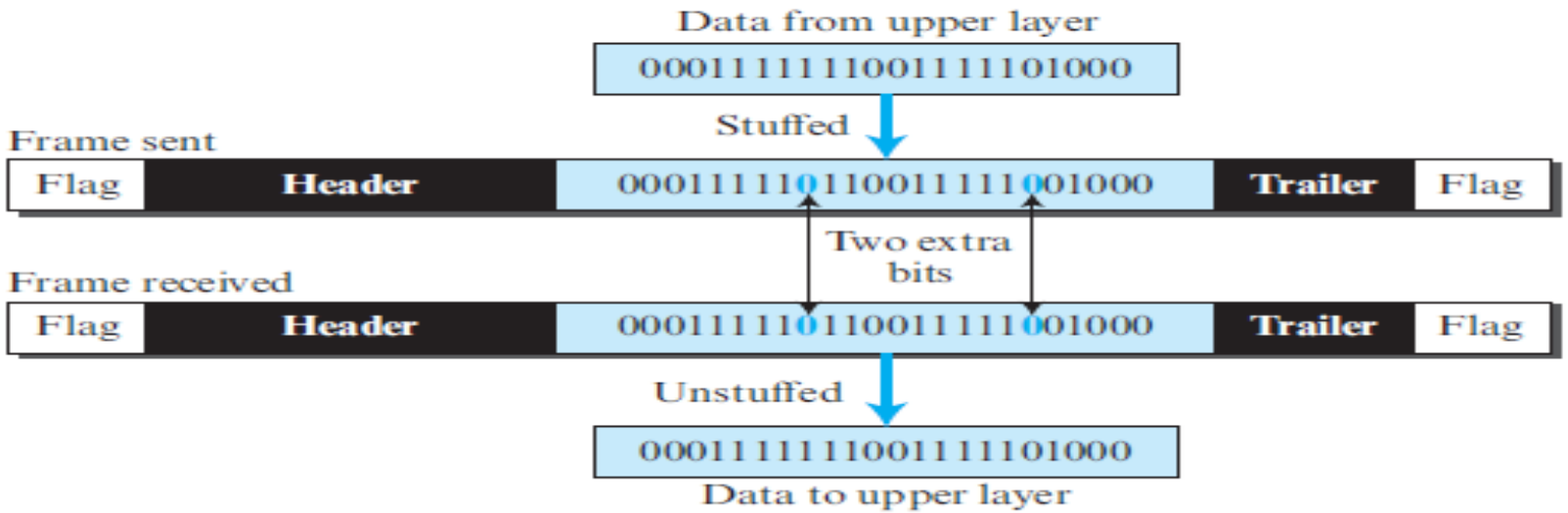


- In addition to headers (and possible trailers), we still need a delimiter to separate one frame from the other.
- Most protocols use a special 8-bit pattern flag, 01111110, as the delimiter to define the beginning and the end of the frame

This flag can create the same type of problem we saw in the character-oriented protocols. That is, if the flag pattern appears in the data, we need to somehow inform the receiver that this is not the end of the frame. We do this by stuffing 1 single bit (instead of 1 byte) to prevent the pattern from looking like a flag. The strategy is called bit stuffing.

Bit stuffing is the process of adding one extra 0 whenever five consecutive 1s follow a 0 in the data, so that the receiver does not mistake the pattern 0111110 for a flag.

Figure 5.7 *Bit stuffing and unstuffing*

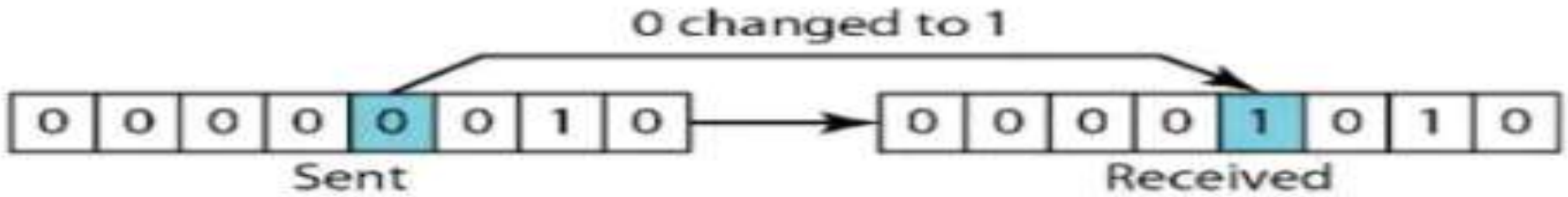


Error detection

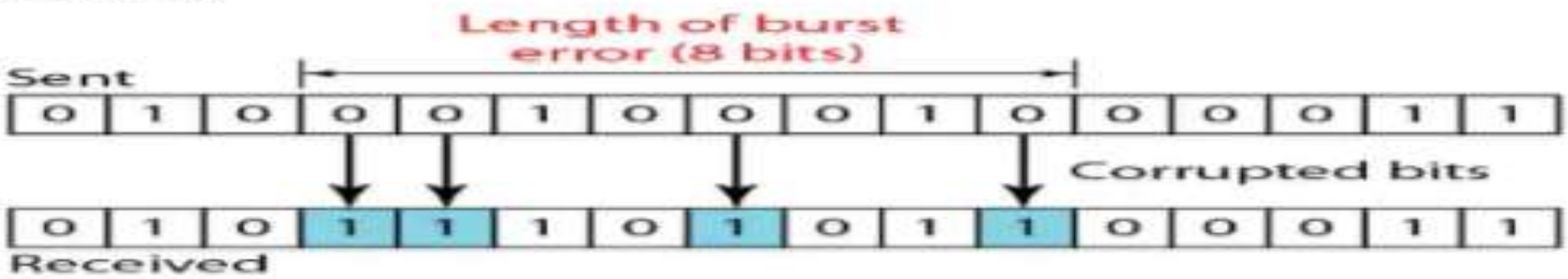
- Bit-level error detection and correction:
  - Between nodes connected by single link
  - Frames with errors handled two ways:
    - Receiving node (tries) to correct the error
    - Receiving node drops the frame

Types of Error Single Bit Errors

In a single bit error, only 1 bit in the data unit changes:



Burst Errors



Error Detection and Correction

- Bits are occasionally flipped in transmission.
  - For example: 1101001 is sent, but 0101011 is received.
- Adding redundancy can allow for detection, and possibly correction, of some errors.
- Simple approach: Repeat each bit
  - Repeat each bit twice. For bit **y**, transmit **yy**. If the receiver gets two different bits, it requests a retransmission.
    - This is an error detecting code.
      - Allows for one error to be detected, but is not error correcting since retransmission is necessary
  - Repeat each bit three times. For each bit **y**, transmit **yyy**.
    - Now the receiver can (most likely) correct a single error.
      - Why?

Problem with the simple approach

- The receiver can detect and correct bit errors if each bit is transmitted three times.
  - How does this affect performance?

- Better approach
  - Parity check codes
    - Has the ability to detect odd number of bit flips using a single parity bit.

Calculating bit string parity

- A bit string has **odd parity** if the number of 1s in the string is odd.
  - 100011, 1, 000010 have odd parity
- A bit string has **even parity** if the number if 1s in the string is even.
  - 01100, 000, 11001001 have even parity
- Assume 0 is an even number

Parity check code

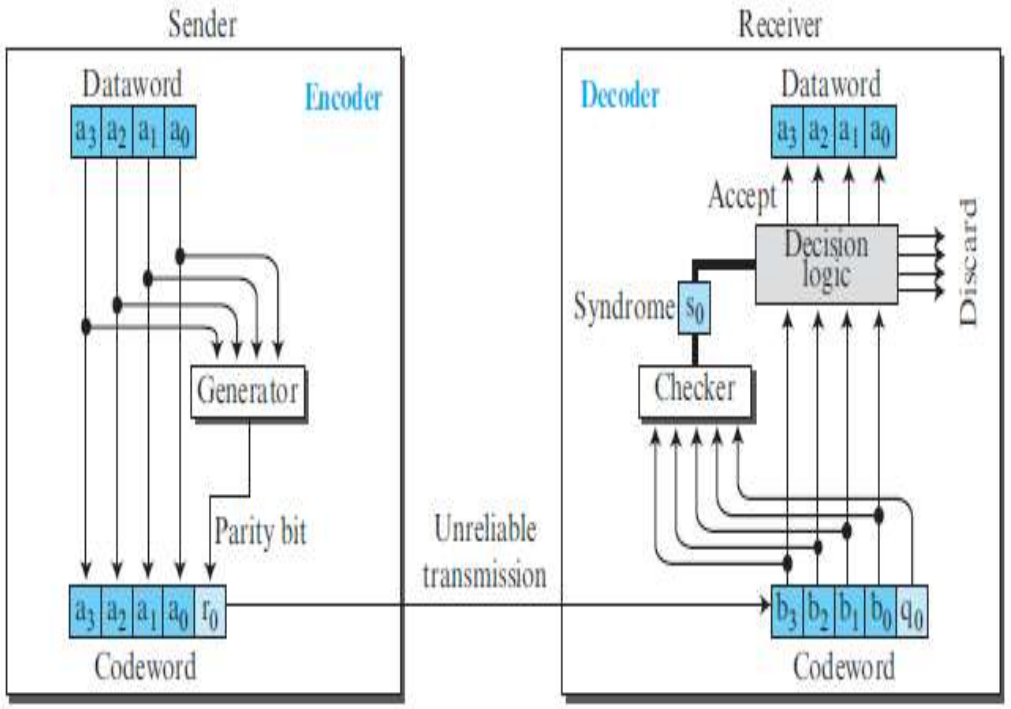
- Assume we are transmitting blocks of *k* bits.
  - A block (w) of length (k) is encoded as (wa), where the value of the **parity bit** (a) is chosen so that (wa) has even parity.

- Example:
  - If w = 10110, we send wa = 101101, which has even parity

  - With no bit flips in the transmission, the receiver gets the bit string exactly as it was sent by the sender. Bit string has even parity.
  - If there are an odd number of bit flips in the transmission, the receiver gets a bit string with odd parity. Retransmission is requested.
  - If there are an even number of bit flips in the transmission, the receiver gets a bit string with even parity. The error(s) go undetected.

  - Another solution?

Figure 5.11 *Encoder and decoder for simple parity-check code*

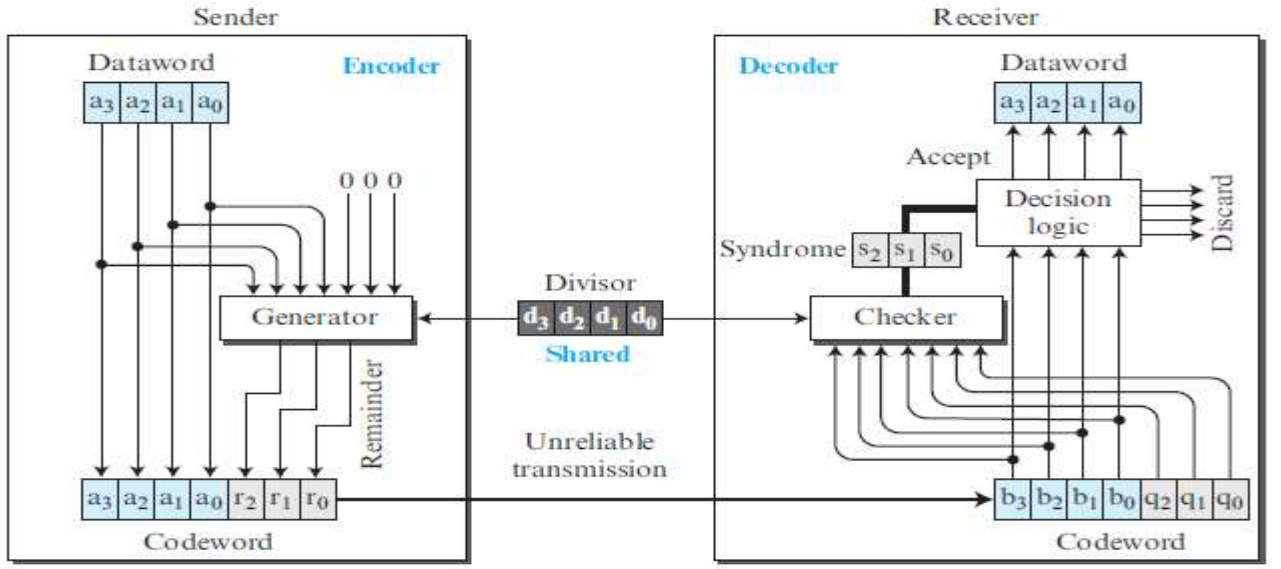


- A parity-check code can detect an odd number of errors

Cyclic Redundancy Check [https://www.youtube.com/watch?v=-oUrtqvUA2o]

- We can create cyclic codes to correct errors.
- A type of cyclic codes called the cyclic redundancy check (CRC) that is used in networks such as LANs and WANs.

Figure 5.12 *CRC encoder and decoder*



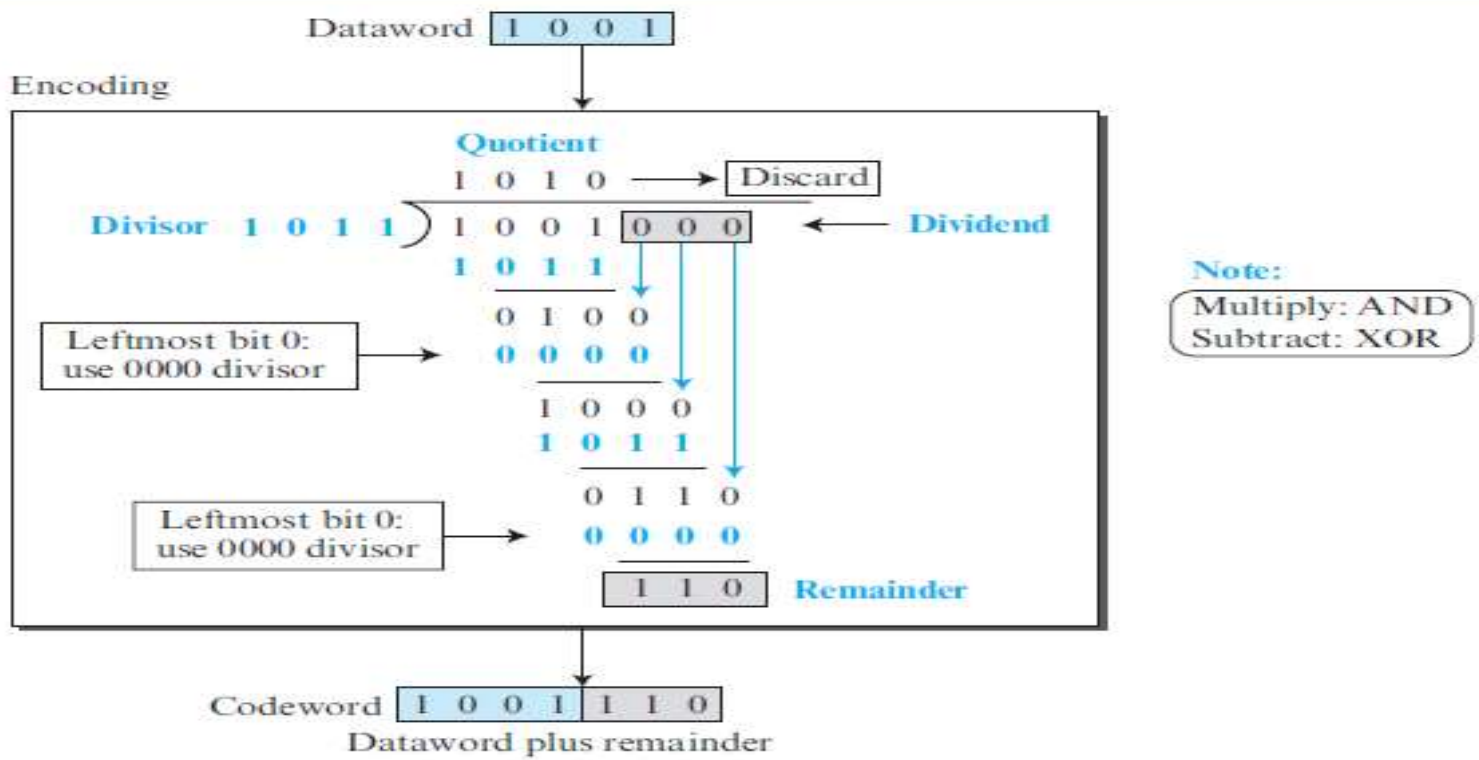
Steps

- dataword has k bits (4 here);
- the codeword has n bits (7 here).
- The size of the dataword is augmented by adding n − k (3 here) 0s 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 (4 here), predefined and agreed upon.
- The generator divides the augmented dataword by the divisor (modulo-2 division).
- The quotient of the division is discarded;
- the remainder (r2r1r0) is appended to the dataword to create the codeword.
- The decoder receives the codeword (possibly corrupted in transition).
- 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 0s, the 4 leftmost bits of the codeword are accepted as the dataword (interpreted as no error);
- otherwise, the 4 bits are discarded (error).

Sender

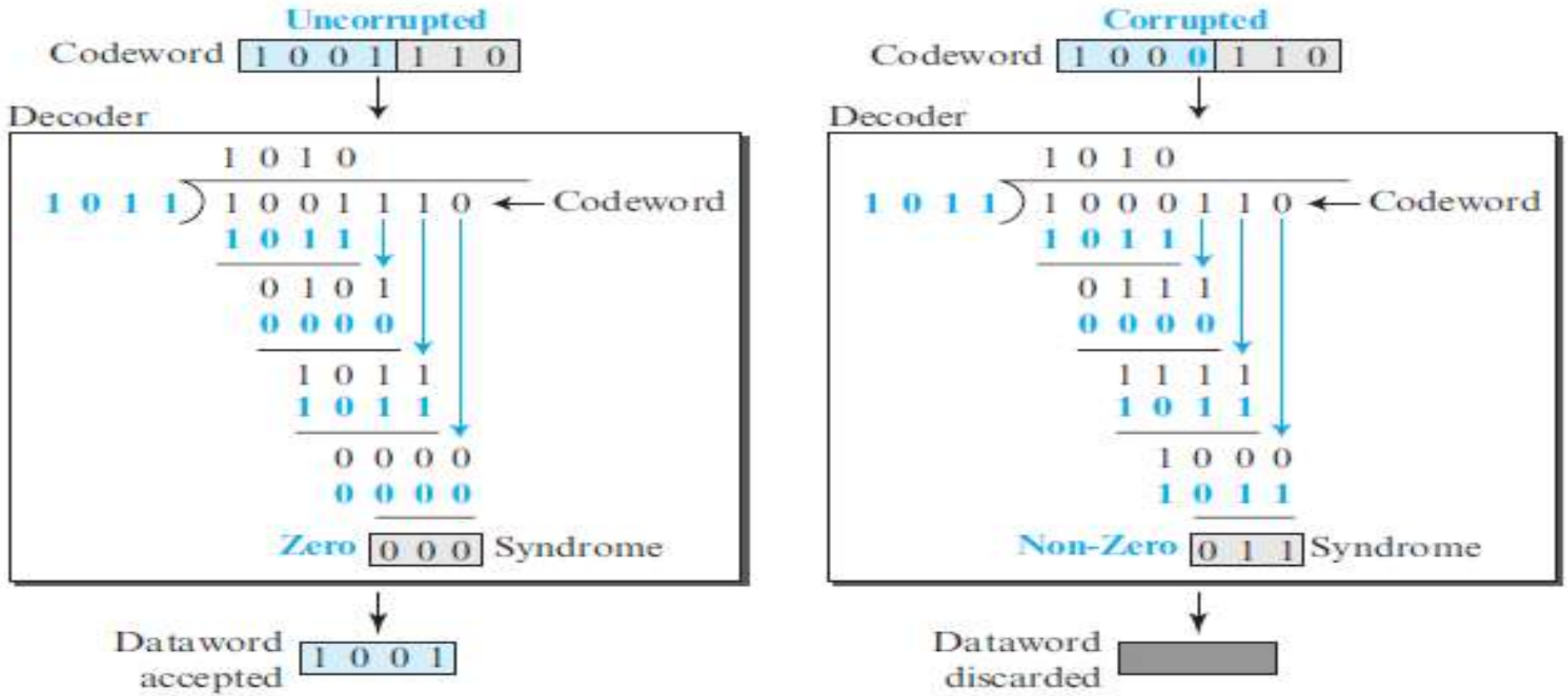
Encoder Let us take a closer look at the encoder. The encoder takes a dataword and augments it with $n - k$ number of 0s. It then divides the augmented dataword by the divisor, as shown in Figure 5.13.

Figure 5.13 *Division in CRC encoder*



Reciver

Figure 5.14 *Division in the CRC decoder for two cases*



Checksum

- Checksum is an error-detecting technique that can be applied to a message of any length.
- In the Internet, the checksum technique is mostly used at the network and transport layer rather than the data-link layer

- At the source, the message is first divided into m-bit units.
- The generator then creates an extra m-bit unit called the checksum, which is sent with the message.
- At the destination, the checker creates a new checksum from the combination of the message and sent checksum.
- If the new checksum is all 0s, the message is accepted; otherwise, the message is discarded .
- Note that in the real implementation, the checksum unit is not necessarily added at the end of the message; it can be inserted in the middle of the message.

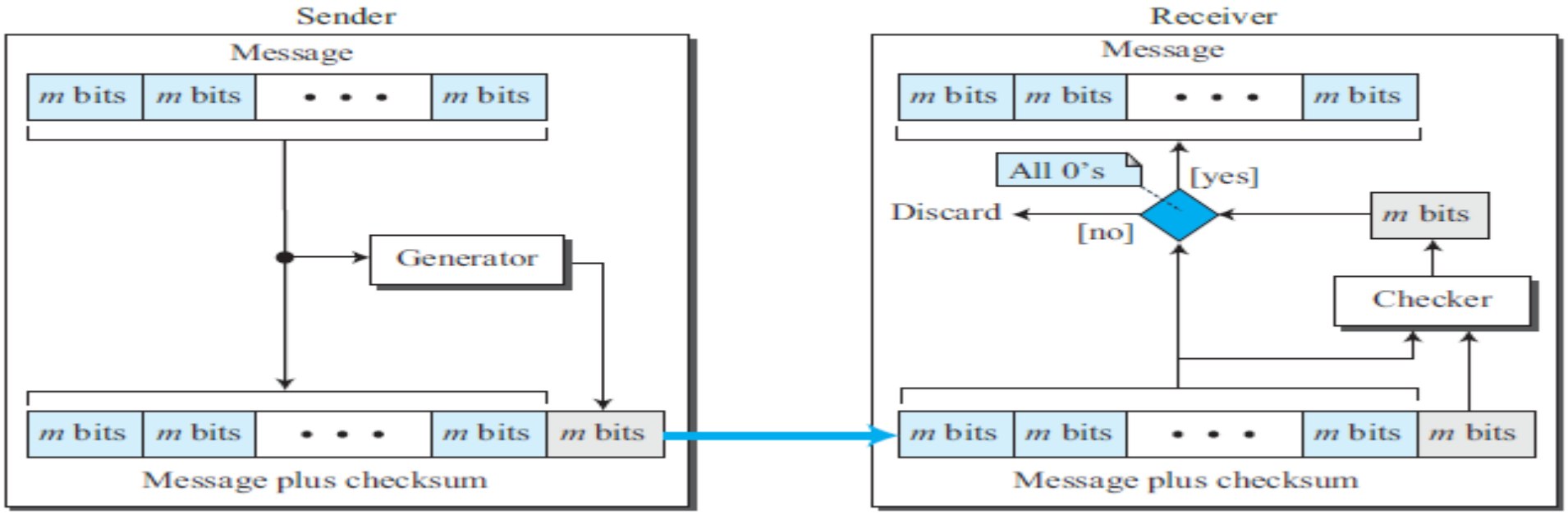
Traditional checksum

- Suppose the message is a list of five 4-bit numbers that we want to send to a destination. In addition to sending these numbers, we send the sum of the numbers. For example, if the set of numbers is (7, 11, 12, 0, 6), we send (7, 11, 12, 0, 6, 36), where 36 is the sum of the original numbers. The receiver adds the five numbers and compares the result with the sum. If the two are the same, the receiver assumes no error, accepts the five numbers, and discards the sum. Otherwise, there is an error somewhere and the message not accepted

One’s Complement Addition

[https://www.youtube.com/watch?v=AtVWnyDDaDI]

Figure 5.15 *Checksum*



Checksum

Checksum = Check + sum.

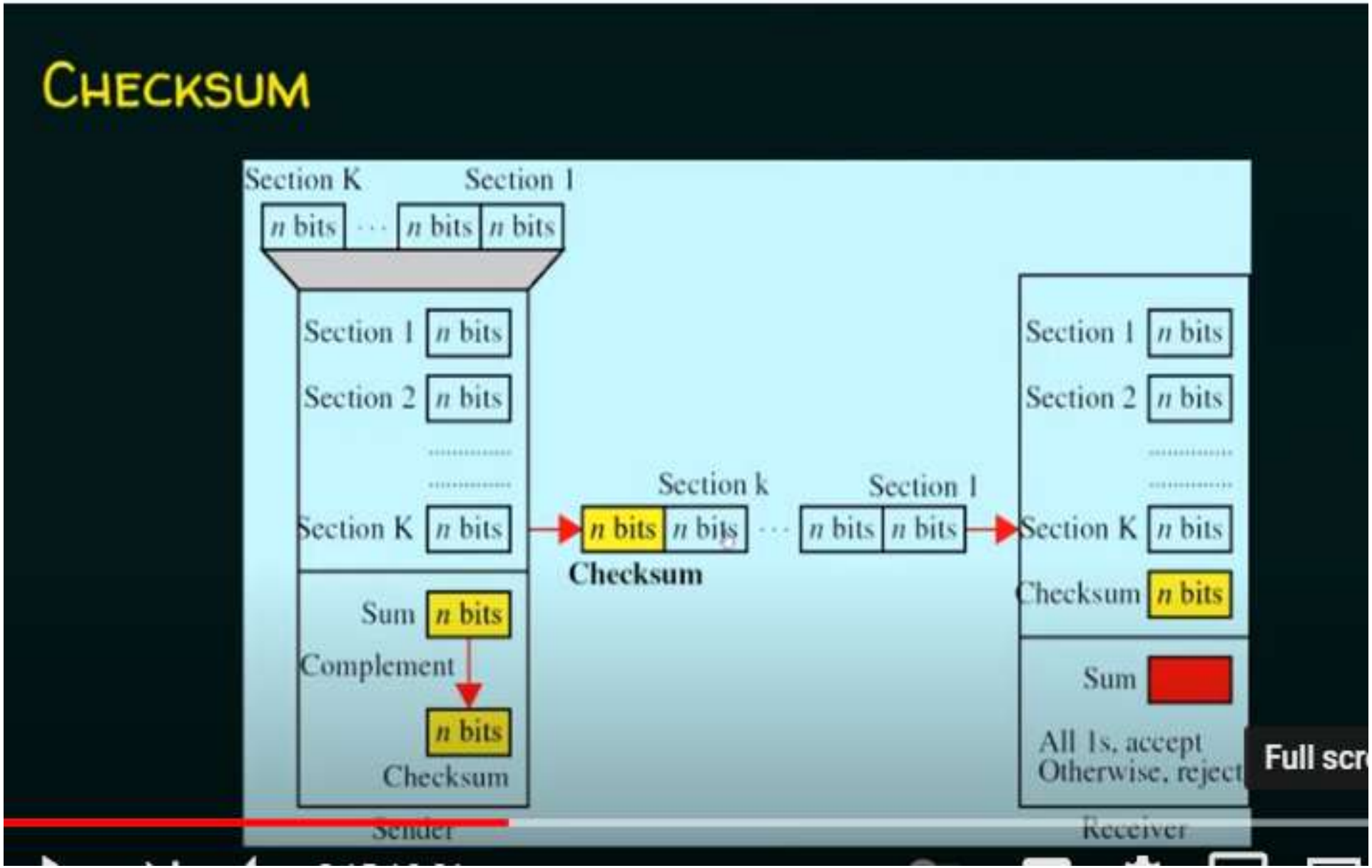
Sender side – Checksum Creation.

Receiver side – Checksum Validation.

Checksum – Operation at sender side

1. Break the original message in to 'k' number of blocks with 'n' bits in each block.
2. Sum all the 'k' data blocks.
3. Add the carry to the sum, if any.
4. Do 1's complement to the sum = Checksum.

Checksum

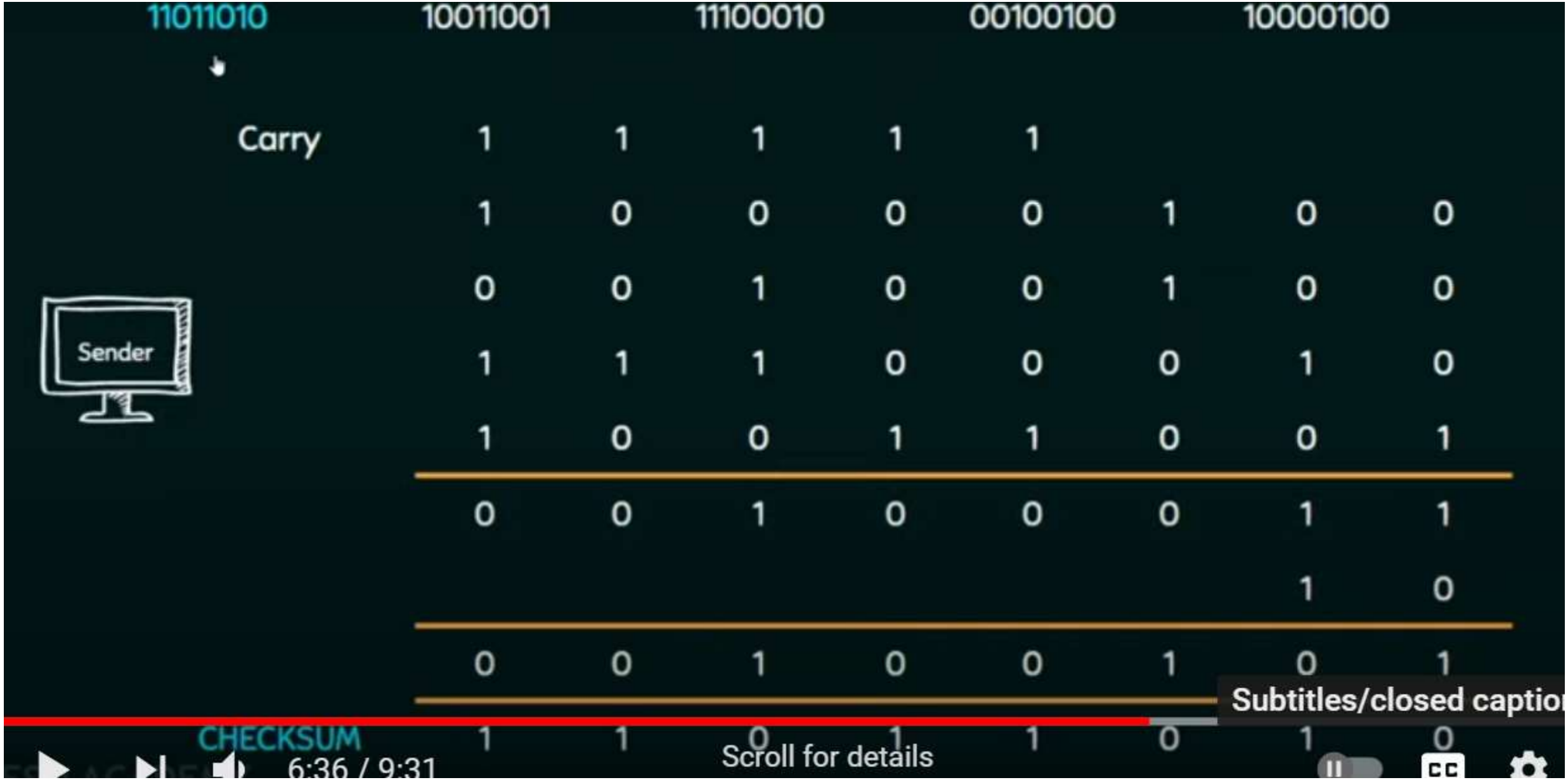

Section K
Section 1
n bits
n bits
n bits
Section 1
n bits
Section 2
n bits
Section K
n bits
Sum
n bits
Complement
n bits
Checksum
Sender
Section k
Section 1
n bits
n bits
n bits
n bits
Checksum
Section 1
n bits
Section 2
n bits
Section K
n bits
Checksum
n bits
Sum
All 1s, accept
Otherwise, reject
Receiver

Checksum – Example

Consider the data unit to be transmitted is:

10011001111000100100100100000100

10011001	11100010	00100100	10000100


11011010
10011001
11100010
00100100
10000100
Carry
1 1 1 1 1
1 0 0 0 0 1 0 0
0 0 1 0 0 1 0 0
1 1 1 0 0 0 1 0
1 0 0 1 1 0 0 1
0 0 1 0 0 0 1 1
1 0
0 0 1 0 0 1 0 1
CHECKSUM 1 1 0 1 1 0 1 0
Sender
Subtitles/closed captio
Scroll for details
6:36 / 9:31

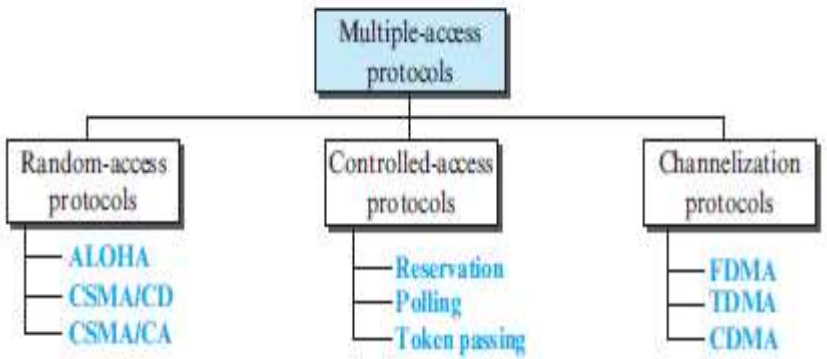
Checksum – Example

	11011010	10011001	11100010	00100100	10000100
Carry	1 1 1 1 1 1				
	1 0 0 0 0 1 0 0				
	0 0 1 0 0 1 0 0				
	1 1 1 0 0 0 1 0				
	1 0 0 1 1 0 0 1				
	1 1 0 1 1 0 1 0				
	1 1 1 1 1 1 0 1				
					1 0
	1 1 1 1 1 1 1 1				

Multiple access protocols (collision and token based)

- data-link layer is divided into two sublayers:
- data link control (DLC)
- media access control (MAC).
- 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

Figure 5.28 *Taxonomy of multiple-access protocols discussed in this chapter*

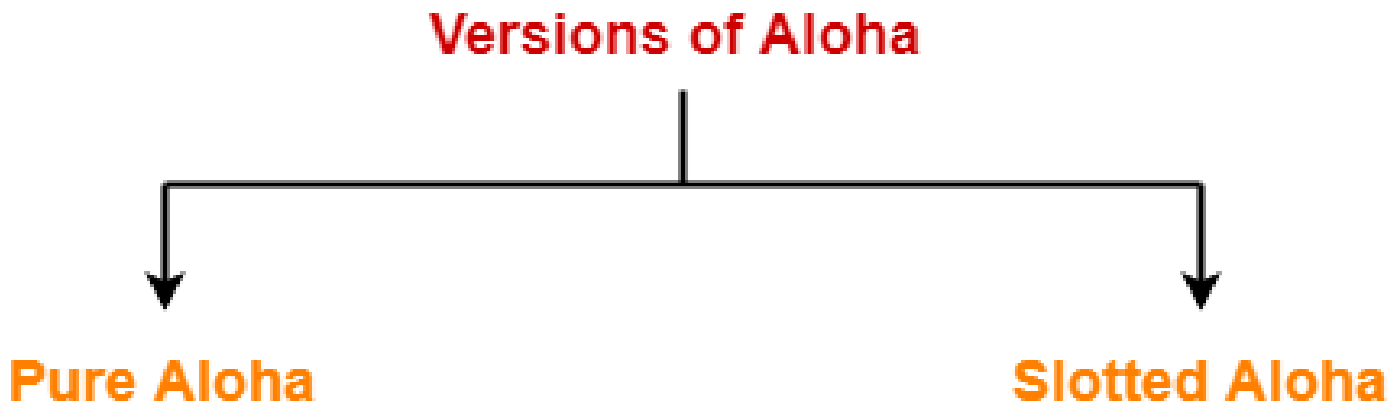


Random Access Protocols

- When node has packet to send
  - transmit at full channel data rate R.
  - no *a priori* coordination among nodes
- two or more transmitting nodes → "collision",
- random access MAC protocol specifies:
  - how to detect collisions
  - how to recover from collisions (e.g., via delayed retransmissions)
- Examples of random access MAC protocols:
  - slotted ALOHA
  - ALOHA
  - CSMA, CSMA/CD, CSMA/CA

Aloha-

There are two different versions of Aloha-



1. Pure Aloha
2. Slotted Aloha

1. Pure Aloha-

- It allows the stations to transmit data at any time whenever they want.
- After transmitting the data packet, station waits for some time.

Then, following 2 cases are possible-

Case-01:

- Transmitting station receives an acknowledgement from the receiving station.
- In this case, transmitting station assumes that the transmission is successful.

Case-02:

- Transmitting station does not receive any acknowledgement within specified time from the receiving station.
- In this case, transmitting station assumes that the transmission is unsuccessful.

Then,

- Transmitting station uses a **Back Off Strategy** and waits for some random amount of time.
- After back off time, it transmits the data packet again.
- It keeps trying until the back off limit is reached after which it aborts the transmission.

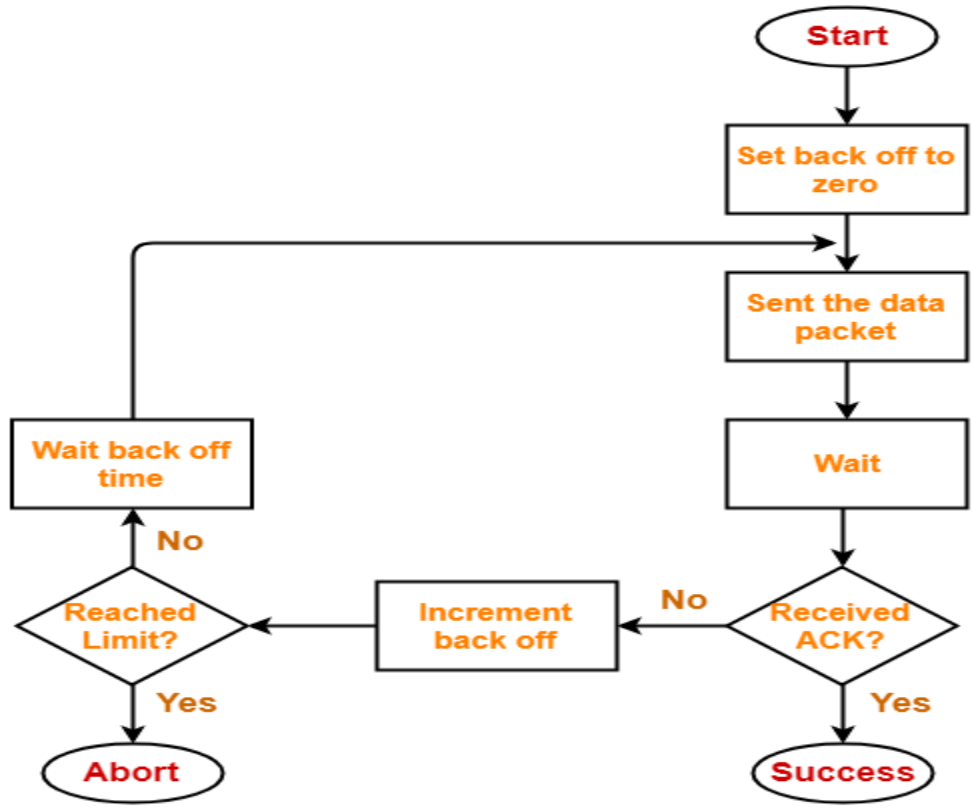
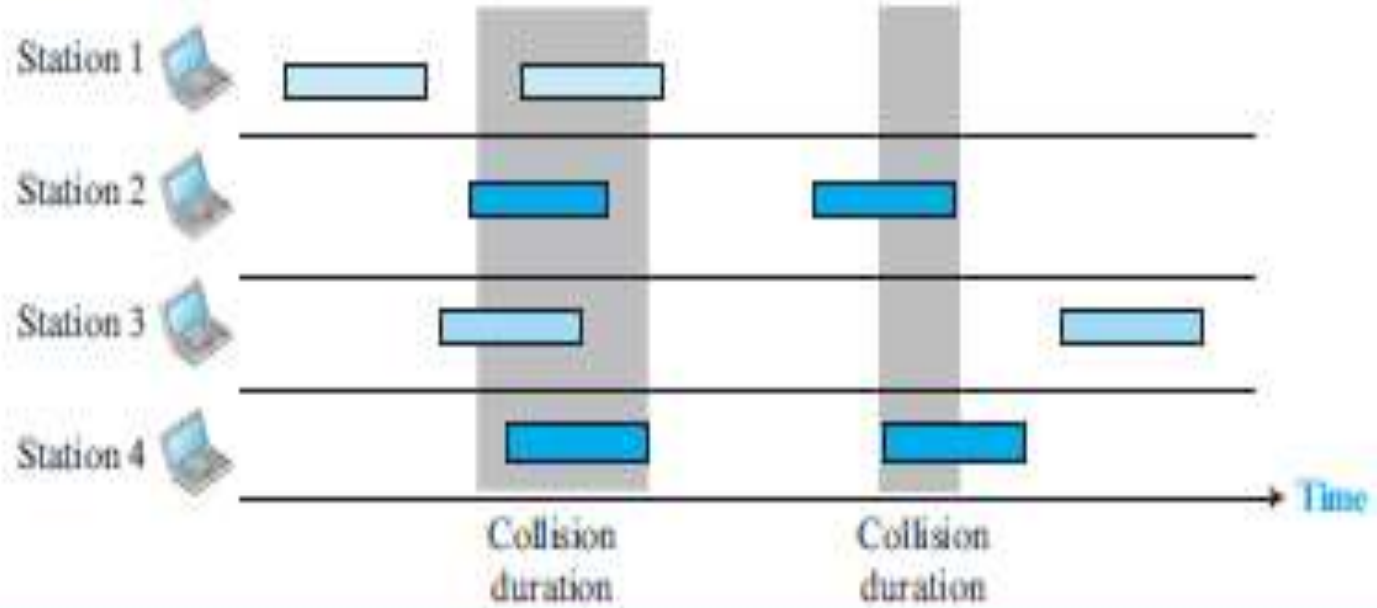


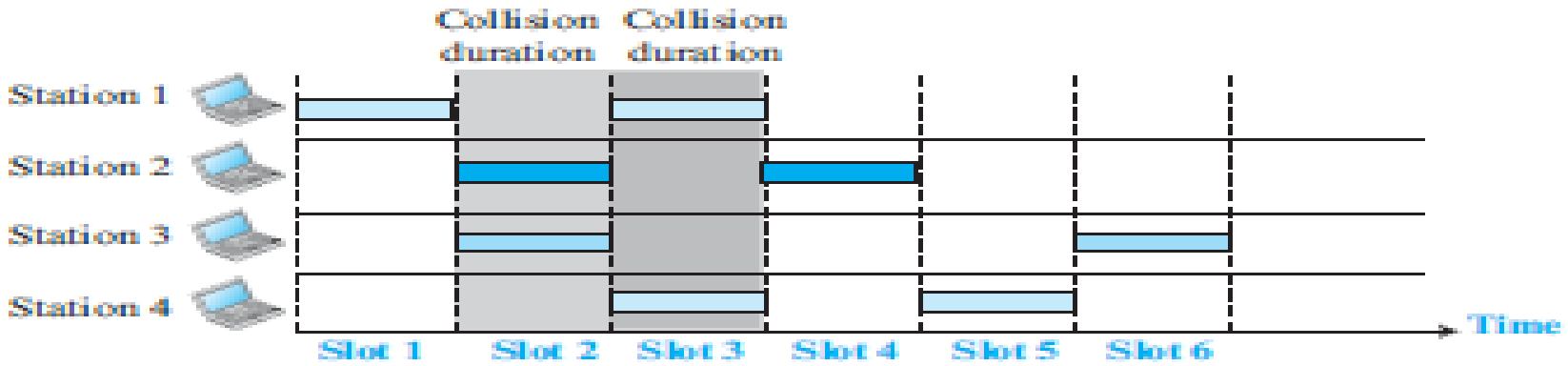
Figure 5.29 *Frames in a pure ALOHA network*



2. Slotted Aloha-

- Slotted Aloha divides the time of shared channel into discrete intervals called as **time slots**.
- Any station can transmit its data in any time slot.
- The only condition is that station must start its transmission from the beginning of the time slot.
- If the beginning of the slot is missed, then station has to wait until the beginning of the next time slot.
- A collision may occur if two or more stations try to transmit data at the beginning of the same time slot.

Figure 5.32 *Frames in a slotted ALOHA network*



Difference Between Pure Aloha And Slotted Aloha-

Pure Aloha	Slotted Aloha
Any station can transmit the data at any time.	Any station can transmit the data at the beginning of any time slot.
The time is continuous and not globally synchronized.	The time is discrete and globally synchronized.
Vulnerable time in which collision may occur $= 2 \times T_t$	Vulnerable time in which collision may occur $= T_t$
Probability of successful transmission of data packet $= G \times e^{-2G}$	Probability of successful transmission of data packet $= G \times e^{-G}$
Maximum efficiency = 18.4% (Occurs at G = 1/2)	Maximum efficiency = 36.8% (Occurs at G = 1)
The main advantage of pure aloha is its simplicity in implementation.	The main advantage of slotted aloha is that it reduces the number of collisions to half and doubles the efficiency of pure aloha.

Problem-

A group of N stations share 100 Kbps slotted ALOHA channel. Each station output a 500 bits frame on an average of 5000 ms even if previous one has not been sent. What is the required value of N?

Solution-

Throughput Of One Station-

Throughput of each station

= Number of bits sent per second

= 500 bits / 5000 ms

= 500 bits / (5000 x 10^{-3} sec)

= 100 bits/sec

Throughput Of Slotted Aloha-

Throughput of slotted aloha

= Efficiency x Bandwidth

= 0.368 x 100 Kbps

= 36.8 Kbps

Total Number Of Stations-

Throughput of slotted aloha = Total number of stations x Throughput of each station

Substituting the values, we get-

36.8 Kbps = N x 100 bits/sec

∴ N = 368

CSMA

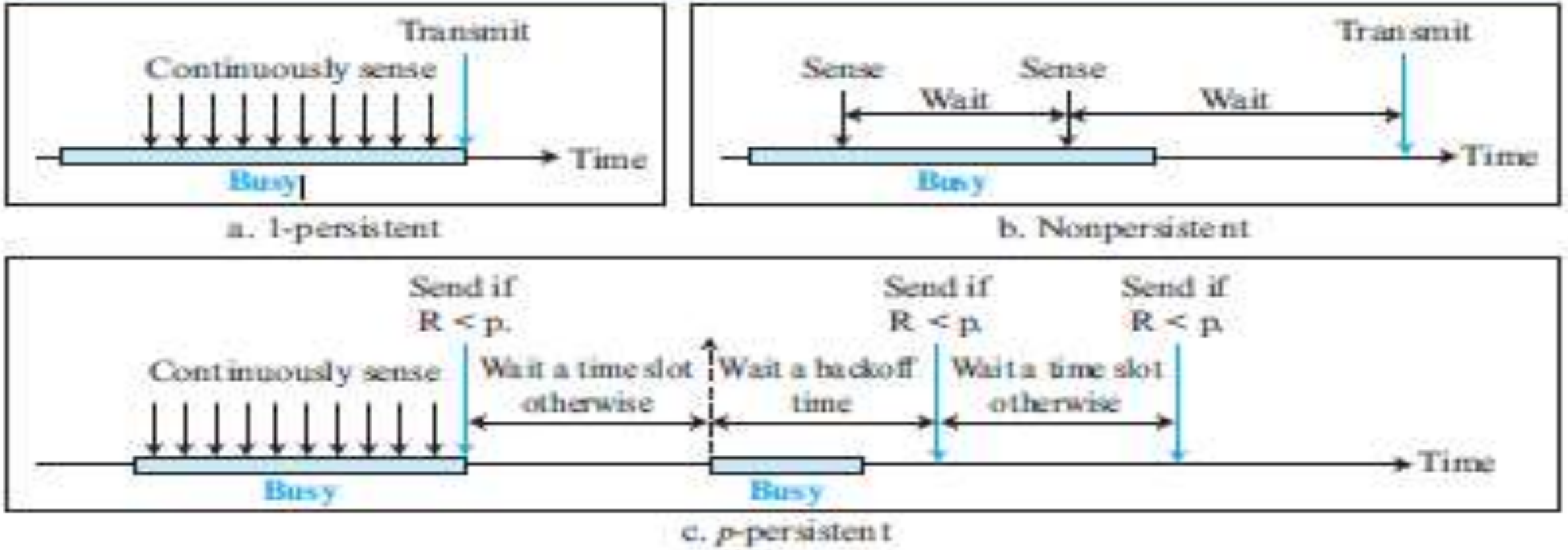
- Carrier sense multiple access (CSMA) requires that each station first listen to the medium (or check the state of the medium) before sending.
- The station is required to first sense the medium (for idle or busy) before transmitting data.
- If it is idle then it sends data
- otherwise it waits till the channel becomes idle.
- However there is still chance of collision in CSMA due to propagation delay.
- *For example, if station A wants to send data, it will first sense the medium.If it finds the channel idle, it will start sending data. However, by the time the first bit of data is transmitted (delayed due to propagation delay) from station A, if station B requests to send data and senses the medium it will also find it idle and will also send data. This will result in collision of data from station A and B.*

CSMA access modes-

- **1-persistent:** The node senses the channel, if idle it sends the data, otherwise it continuously keeps on checking the medium for being idle and transmits unconditionally(with 1 probability) as soon as the channel gets idle.
- **Non-Persistent:** The node senses the channel, if idle it sends the data, otherwise it checks the medium after a random amount of time (not continuously) and transmits when found idle.
- **P-persistent:** The node senses the medium, if idle it sends the data with p probability. If the data is not transmitted ((1-p) probability) then it waits for some time and checks the medium again, now if it is found idle then it send with p probability. This repeat continues until the frame is sent. It is used in Wifi and packet radio systems.
- **O-persistent:** Superiority of nodes is decided beforehand and transmission occurs in that order. If the medium is idle, node waits for its time slot to send data.

Persistence Methods

Figure 5.36 *Behavior of three persistence methods*



Contention Protocols (Cont'd)

- **CSMA** (Carrier Sense Multiple Access)
  - Improvement: Start transmission only if no transmission is ongoing
- **CSMA/CD** (CSMA with Collision Detection)
  - Improvement: Stop ongoing transmission if a collision is detected
- **CSMA/CA** (CSMA with Collision Avoidance)
  - Improvement: Wait a random time and try again when carrier is quiet. If still quiet, then transmit
- **CSMA/CA with ACK**
- **CSMA/CA with RTS/CTS**

Description of CSMA/CD

1. If the medium is idle, transmit; otherwise, go to step 2.

2. If the medium is busy, continue to listen until the channel is idle, then transmit immediately.

3. If a collision is detected during transmission, transmit a brief jamming signal to assure that all stations know that there has been a collision and then cease transmission.

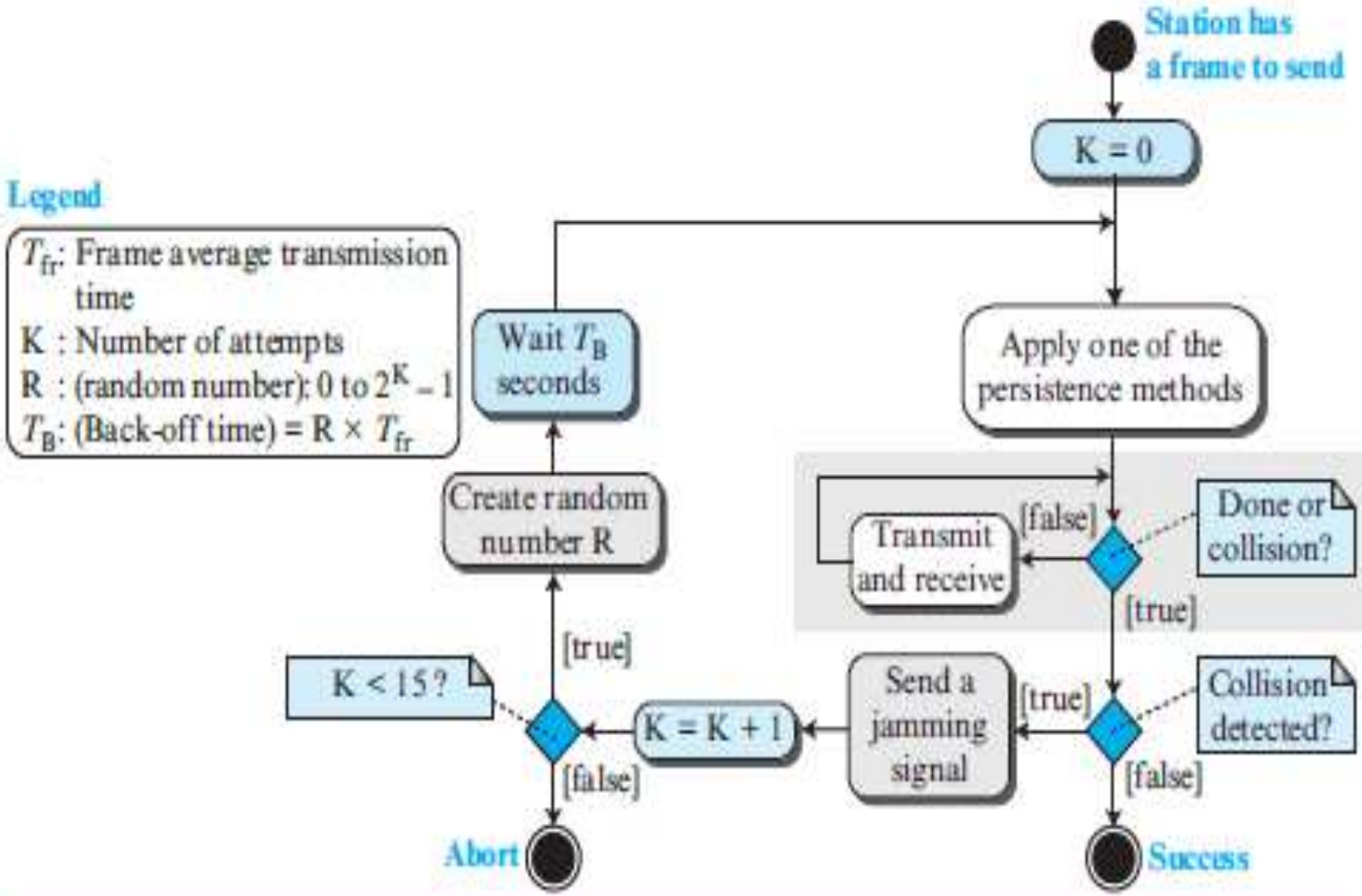
4. After transmitting the jamming signal, wait a random amount of time, referred to as the **backoff**, then attempt to transmit again (repeat from step 1).

CSMA/CD (CSMA with Collision Detection)

- In CSMA, if 2 terminals begin sending packet at the same time, each will transmit its complete packet (although collision is taking place).
- Wasting medium for an entire packet time.
- CSMA/CD

Step 1:	If the medium is idle, transmit
Step 2:	If the medium is busy, continue to listen until the channel is idle then transmit
Step 3:	If a collision is detected during transmission, cease transmitting
Step 4:	Wait a random amount of time and repeats the same algorithm

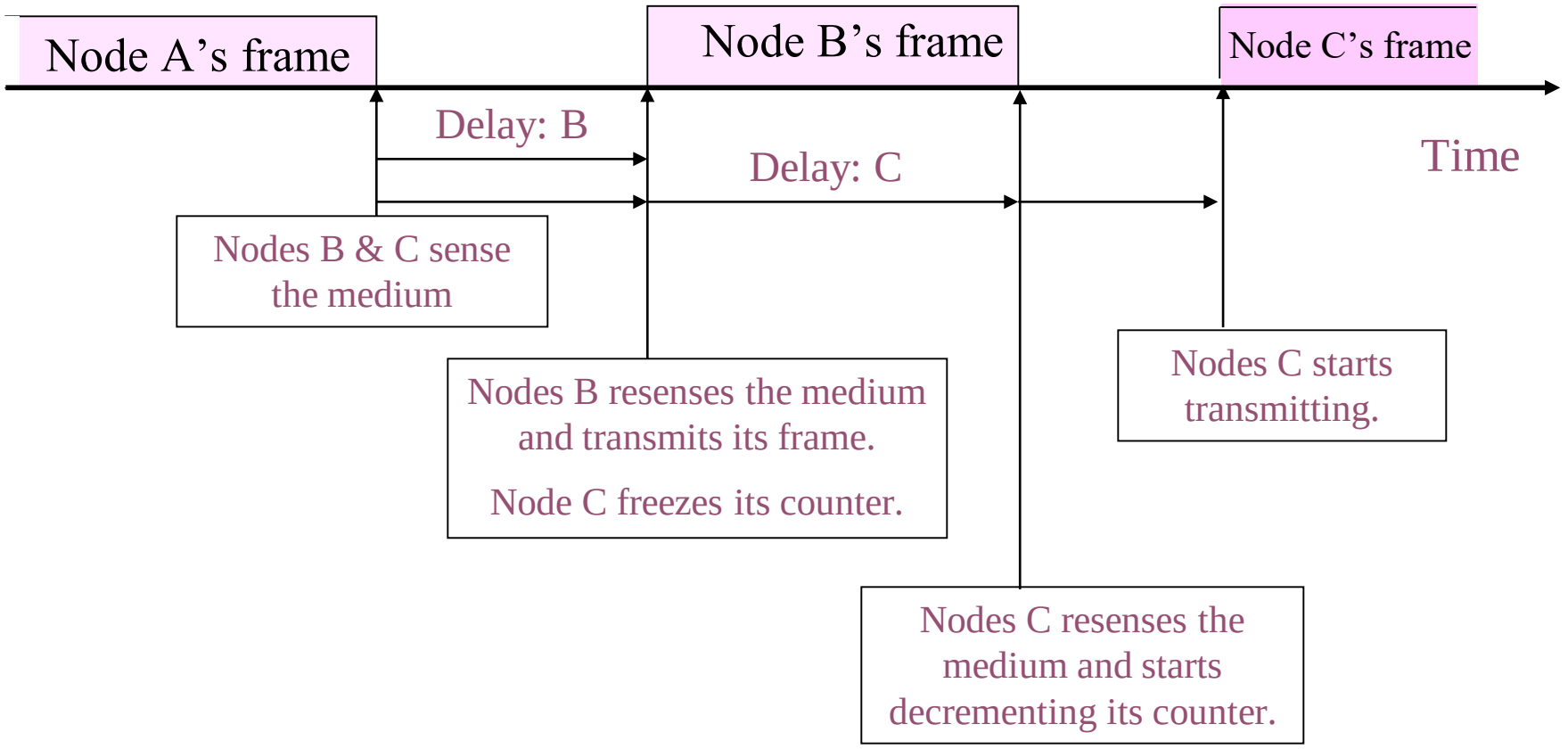
Figure 5.40 *Flow diagram for the CSMA/CD*



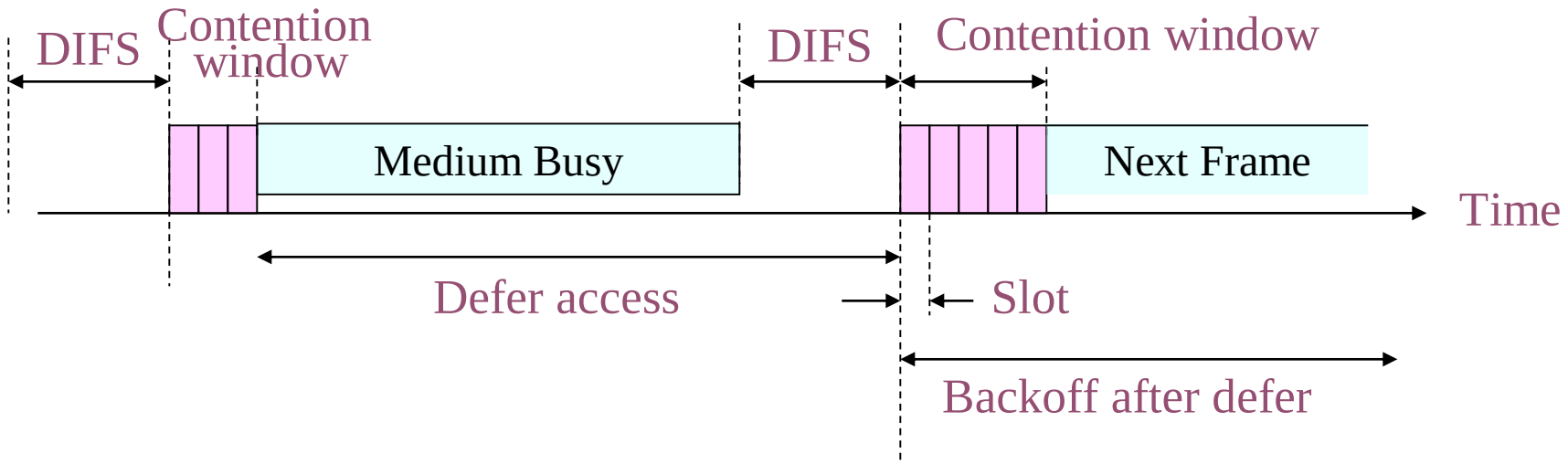
CSMA/CA (CSMA with collision Avoidance)

- All terminals listen to the same medium as CSMA/CD.
- Terminal ready to transmit senses the medium.
- If medium is busy it waits until the end of current transmission.
- It again waits for an additional predetermined time period DIFS (Distributed inter frame Space).
- Then picks up a random number of slots (the initial value of backoff counter) within a contention window to wait before transmitting its frame.
- If there are transmissions by other terminals during this time period (backoff time), the terminal freezes its counter.
- It resumes count down after other terminals finish transmission + DIFS. The terminal can start its transmission when the counter reaches to zero.

CSMA/CA (Cont'd)



CSMA/CA Explained

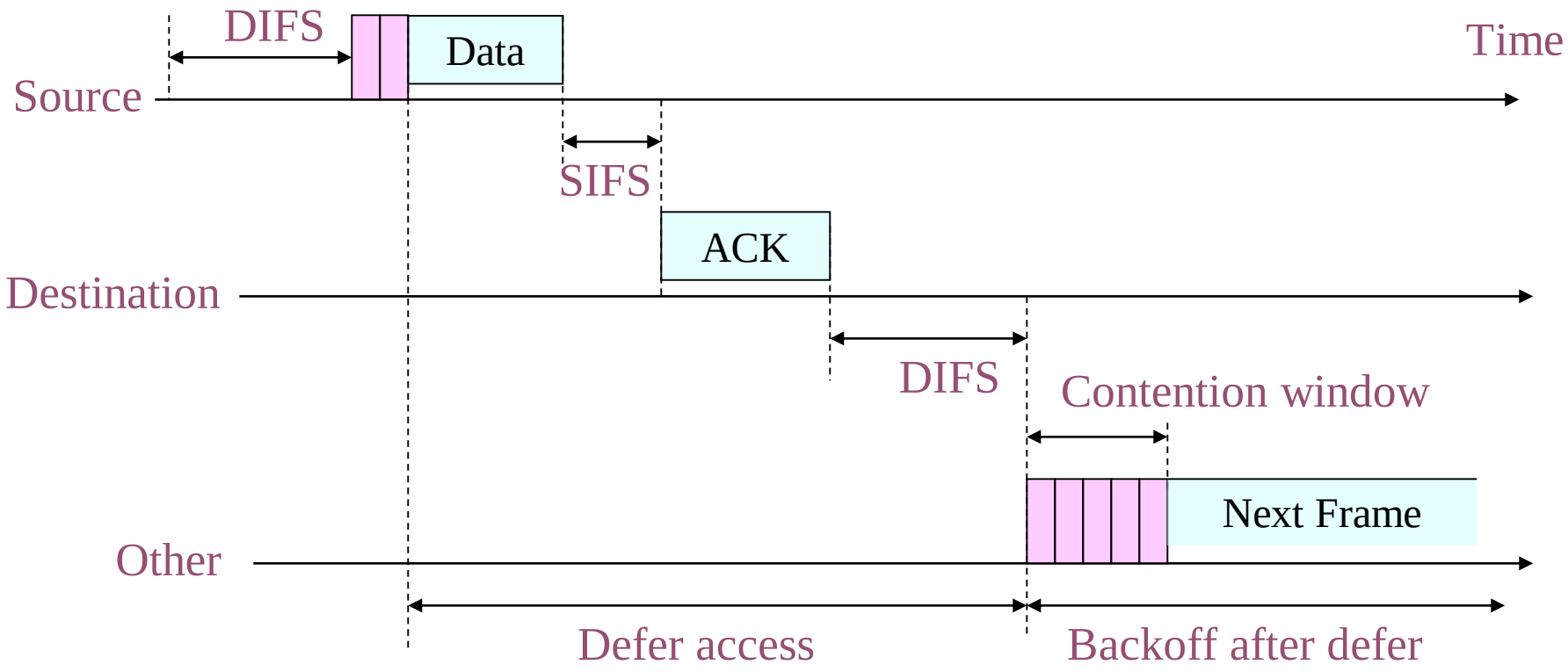


DIFS – Distributed Inter Frame Spacing

CSMA/CA with ACK

- Immediate Acknowledgements from receiver upon reception of data frame without any need for sensing the medium.
- ACK frame transmitted after time interval SIFS (*Short Inter-Frame Space*) (*SIFS < DIFS*)
- Receiver transmits ACK without sensing the medium.
- If ACK is lost, retransmission done.

CSMA/CA/ACK

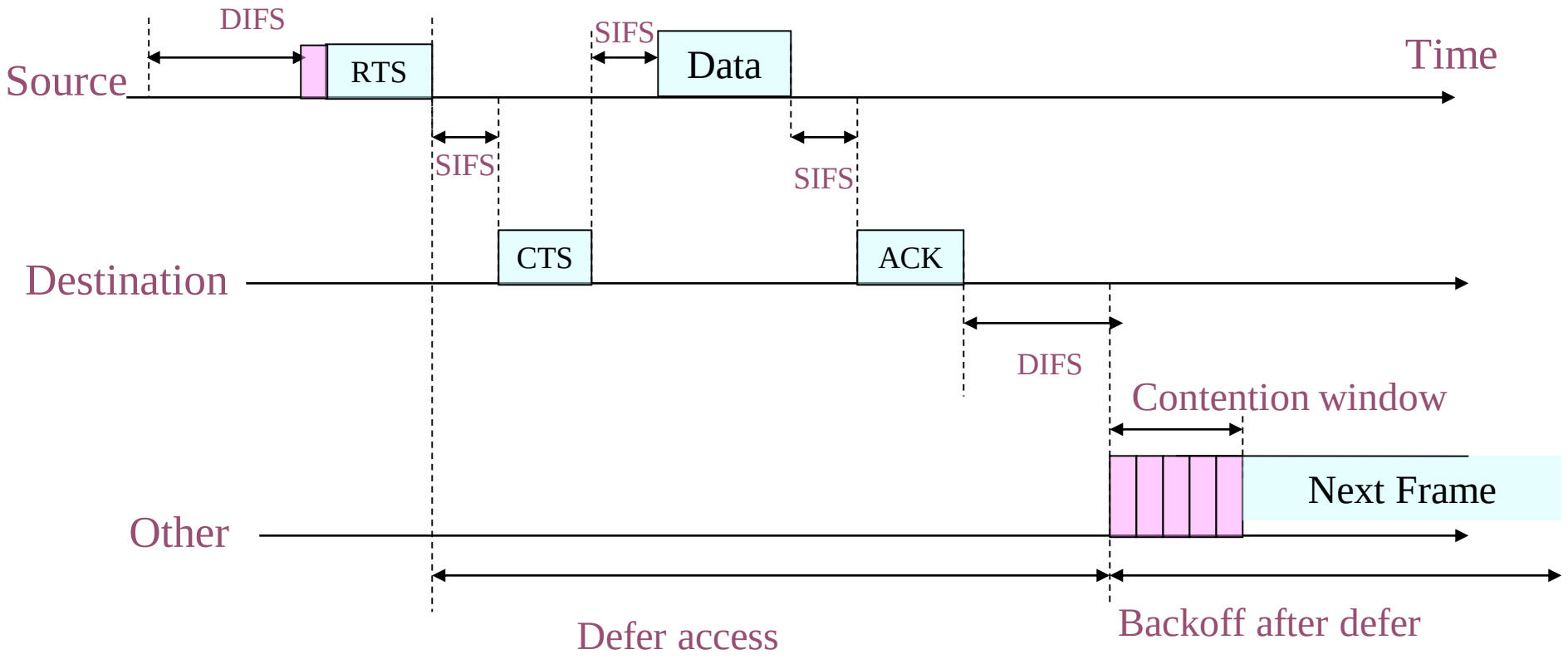


SIFS – Short Inter Frame Spacing

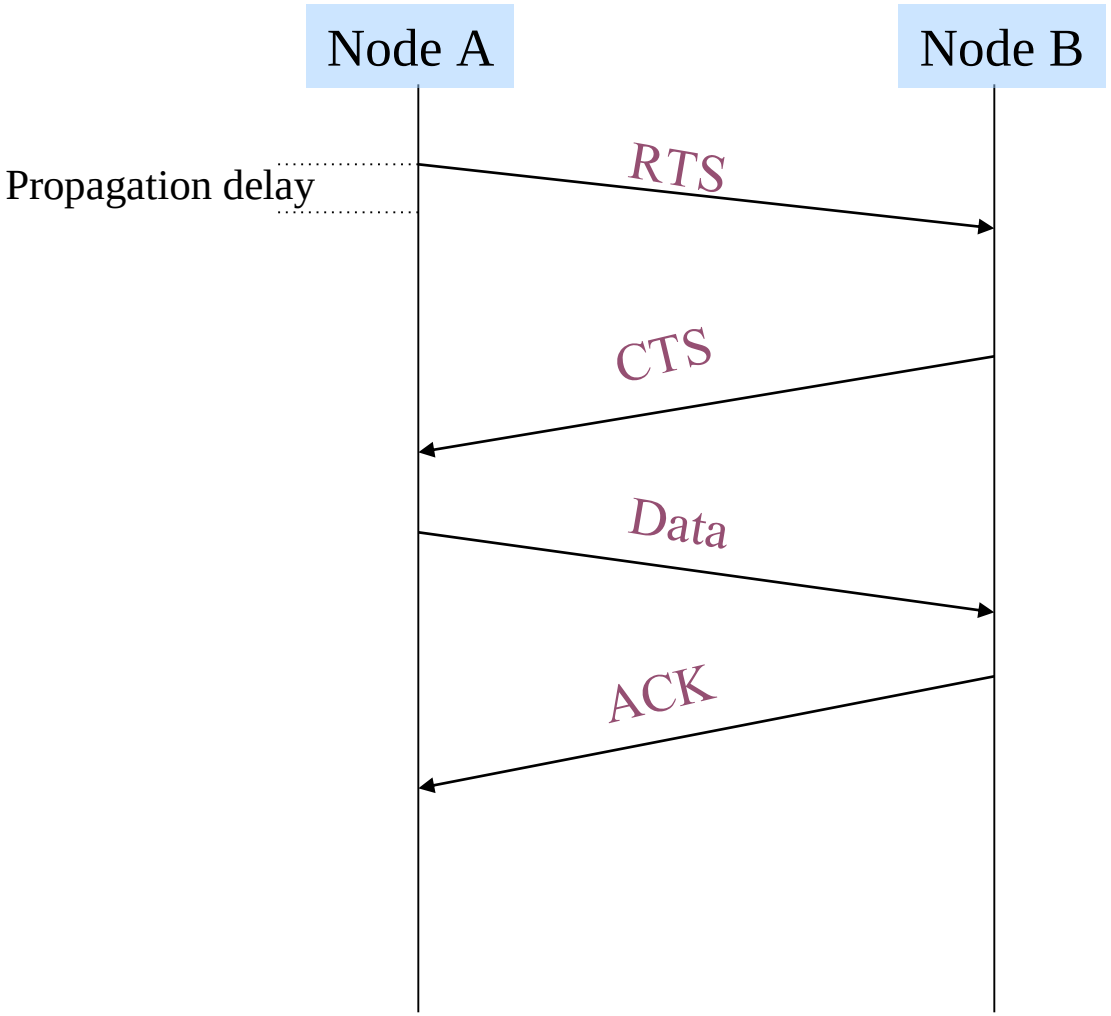
CSMA/CA with RTS/CTS

- Transmitter sends an RTS (request to send) after medium has been idle for time interval more than DIFS.
- Receiver responds with CTS (clear to send) after medium has been idle for SIFS.
- Then Data is exchanged.
- RTS/CTS is used for reserving channel for data transmission so that the collision can only occur in control message.

CSMA/CA with RTS/CTS (Cont'd)



RTS/CTS



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. We discuss three popular controlled-access methods.

Reservation
Polling
Token Passing

RESERVATION

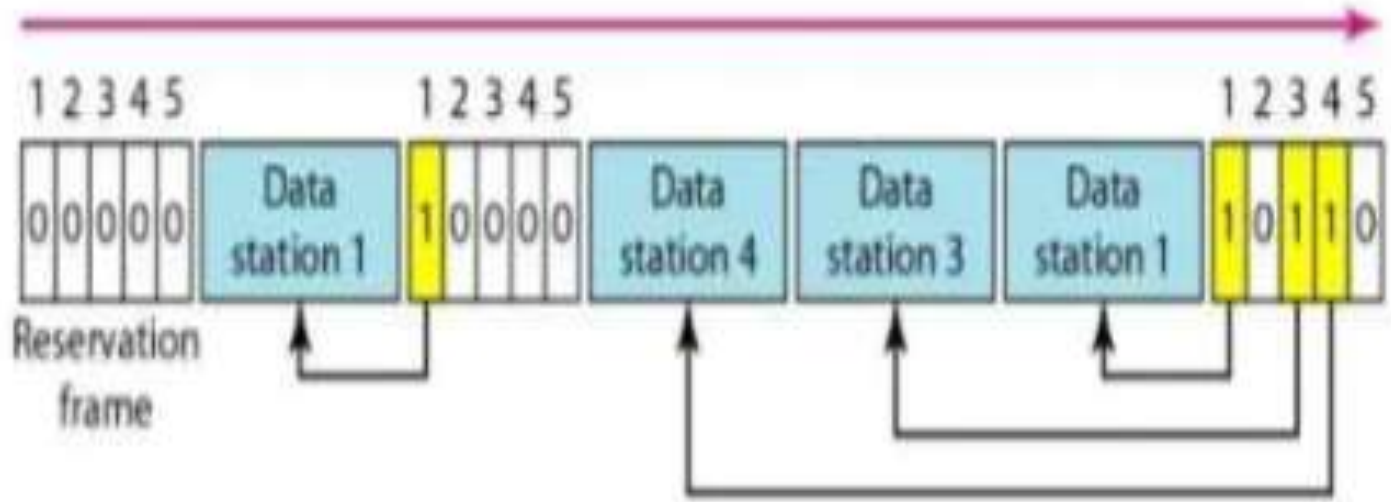
- In the reservation method, a station needs to make a reservation before sending data.
- Time is divided into intervals.
- In each interval, a reservation frame precedes the data frames sent in that interval

RESERVATION

- 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

RESERVATION

- a situation with five stations and a five minislot reservation frame.



Polling

- To impose order on a network of independent users and to establish one station in the network as a controller that periodically polls all other stations which is called ***Polling***.

- There are two general polling policies:
  i. Round Robin Order
  ii. Priority Order

- It works with topologies in which one device is designated as a ***Primary*** Station and the other devices are ***Secondary*** Stations.

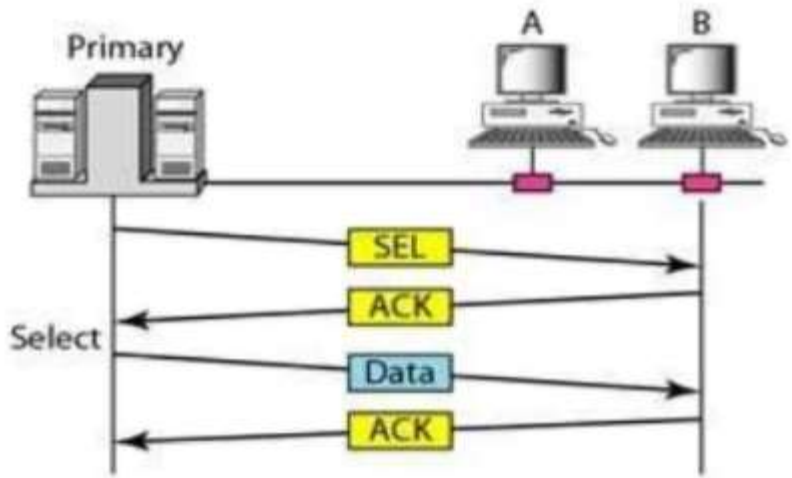
- The Primary device controls the link, where as the secondary follows it's instructions.

- Exchange of data must be made through the primary device even though the final destination is secondary.

SELECT FUNCTION:

- Whenever primary has something to send, it sends the message to each node.
- Before Sending the data, it creates and transmits a Select(*SEL*) frame, one field of it includes the address of the intended secondary.
- While sending, the primary should know whether the target device is ready to receive or not.

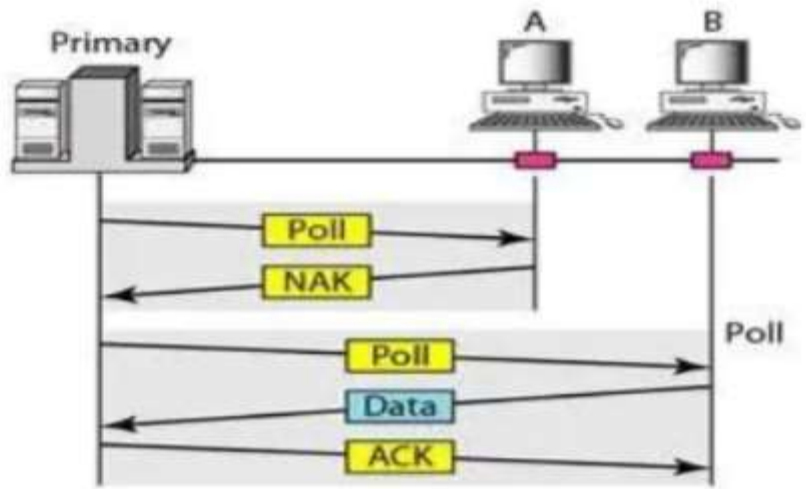
- Hence, it alerts the secondary for the upcoming transmission and wait for an acknowledgement (*ACK*) of secondary's status.



POLL FUNCTION:

- When the primary is ready to receive data, it must ask (*poll*) each device if it has anything to send.
- If the secondary has data to transmit, it sends the data frame. Otherwise, it sends a negative acknowledgement(*NAK*) .
- The primary then polls the next secondary. When the response is positive (a data frame), the primary reads the frame and returns an acknowledgment (*ACK*).

- There are two possibilities to terminate the transmission: either the secondary sends all data, finishing with an *EOT* frame, or the primary says timer is up.



Advantages:

- Priorities can be assigned to ensure faster access from some secondary .
- Maximum and minimum access times and data rates on the channel are predictable and fixed.

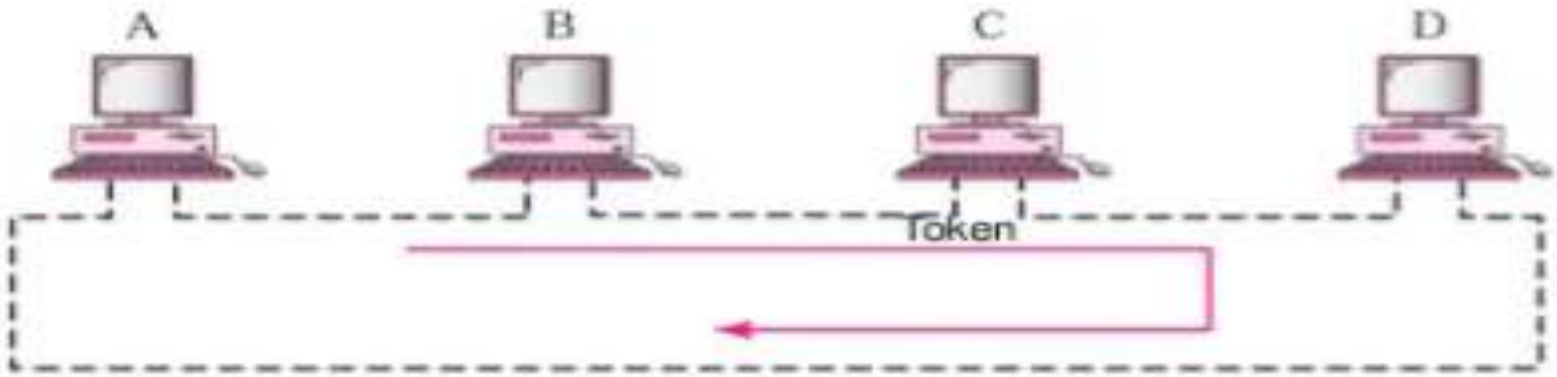
Drawbacks:

- High dependence on the reliability of the controller.
- Increase in turn around time reduces the channel data rate under low loads and it's throughput.

Token passing

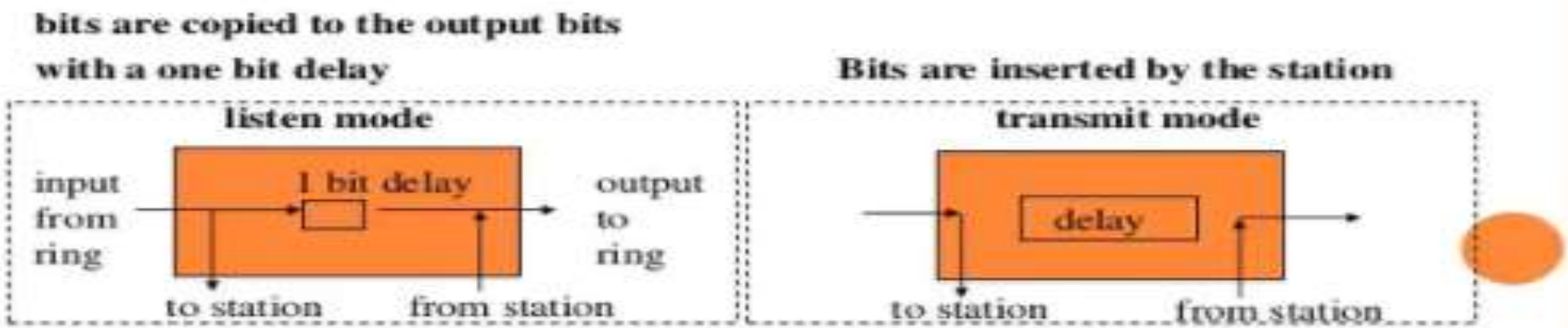
FEATURES:

- A Station is authorized to send data when it receives a special frame called a Token.
- Stations are arranged around a ring (physically or logically)
  - A Token circulates around a ring
- If a station needs to send data ,it waits for the token
- The Station captures the token and sends one or more frames as long as the allocated time has not expired
- It releases the token to be used by the successor station.

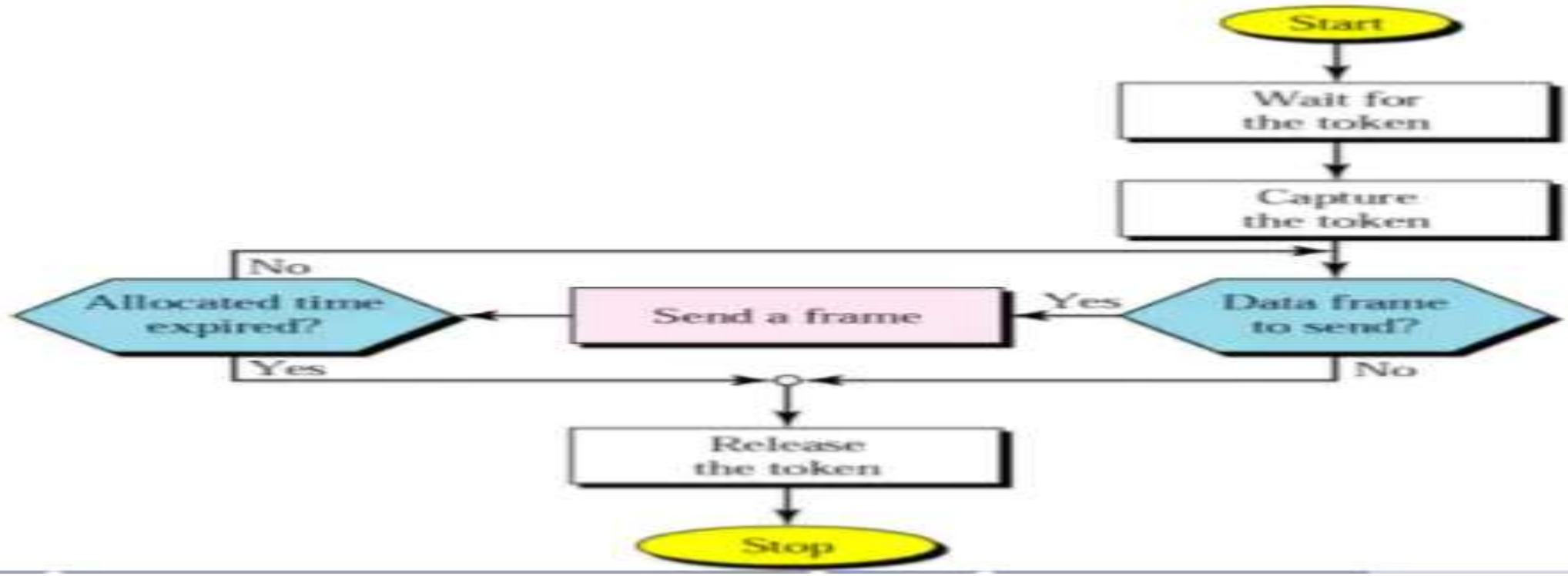


Station Interface is in two states :

- **Listen state**: Listen to the arriving bits and check the destination address to see if it is its own address. If yes the frame is copied to the station otherwise it is passed through the output port to the next station.
- **Transmit state**: station captures a special frame called **free token** and transmits its frames. **Sending** station is responsible for **reinserting** the free token into the ring medium and for **removing** the transmitted frame from the medium.



TOKEN PASSING FLOW CHART :



Token Management :

- We need token management , if there is a loss of token or it is destroyed when a station fails
- We can assign priorities as which station can receive the token.

Network Topology :

- The way in which different systems and nodes are connected and communicate with each other is determined by topology of the network.

Topology can be physical or logical.

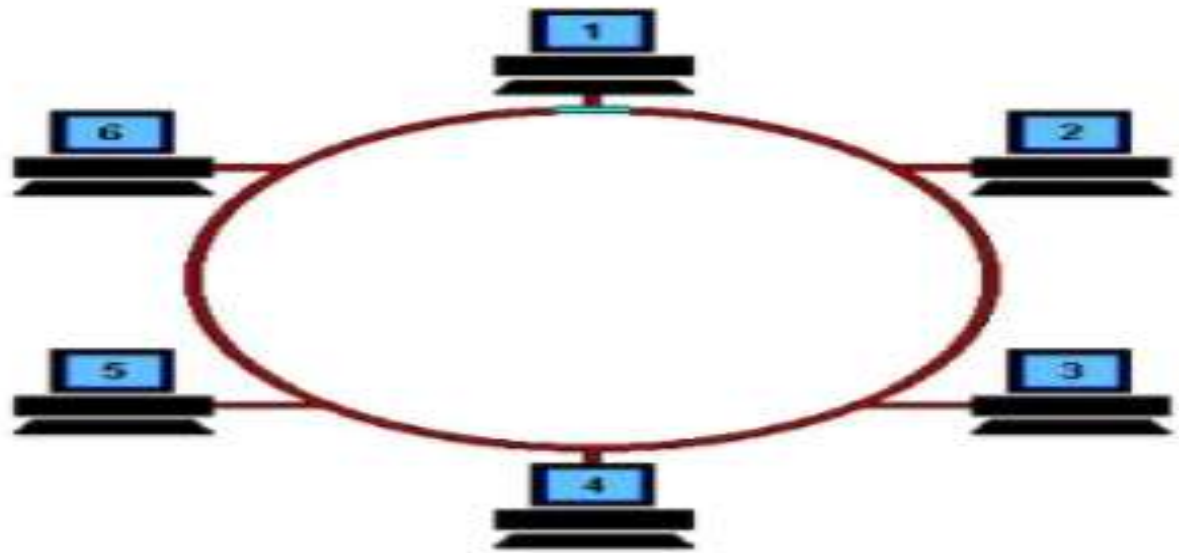
Physical Topology is the physical layout of nodes, workstations and cables in the network

logical topology is the way information flows between different components.

TYPES OF LOGICAL RINGS :

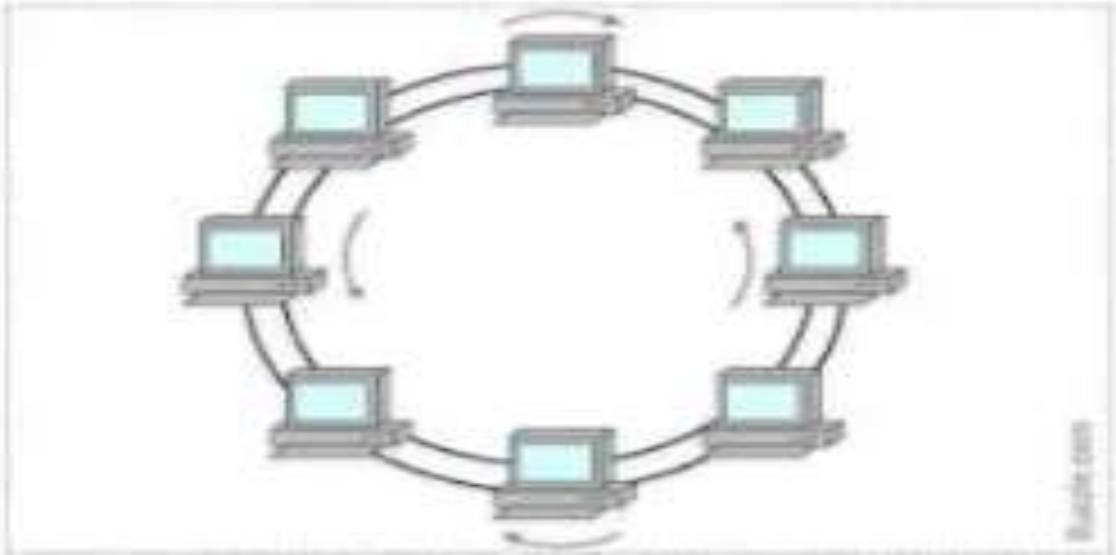
- Physical Ring Topology

  The way that the workstations are connected to the network through the actual cables that transmit data is same as the physical structure of the network .



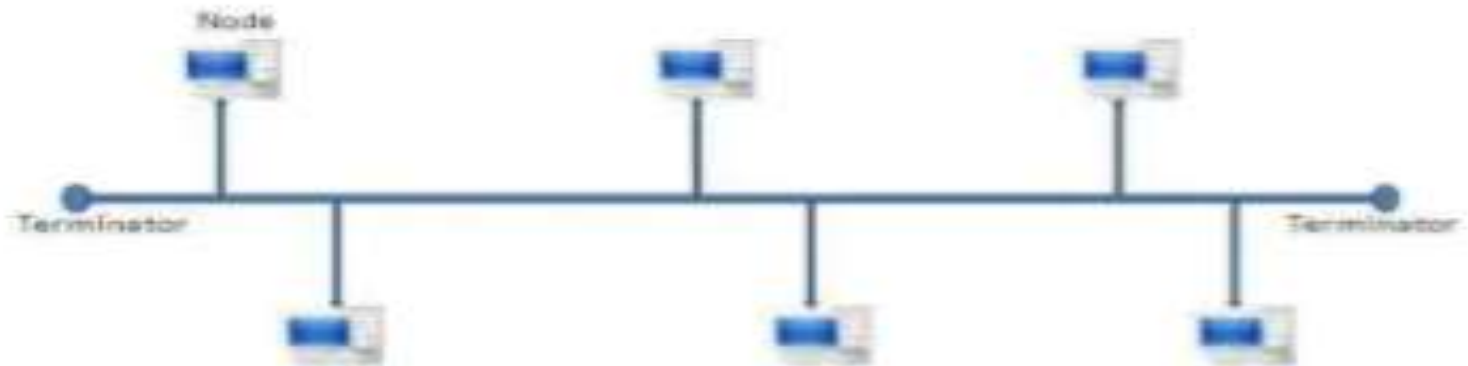
DUAL RING TOPOLOGY :

- A network topology in which two concentric rings connect each node on a network
- Typically, the secondary ring in a dual-ring topology is redundant.
- It is used as a backup in case the primary ring fails.
- In these configurations, data moves in opposite directions around the rings.



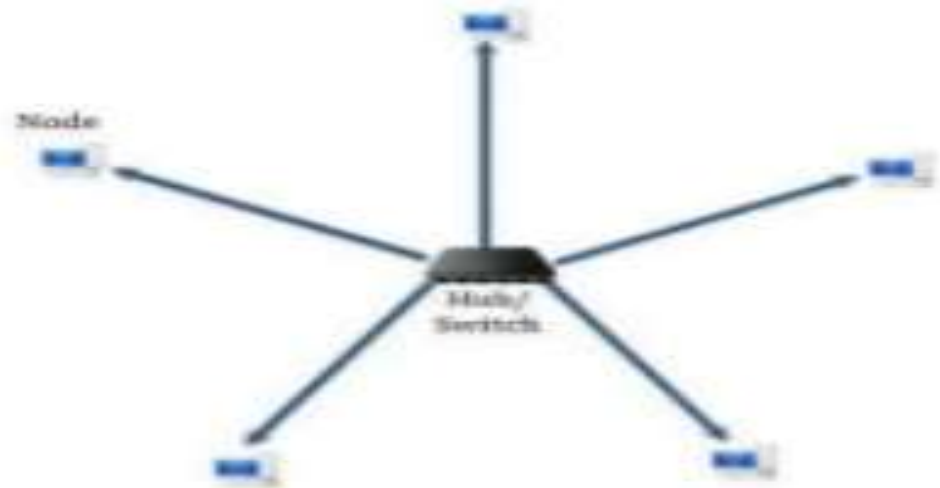
BUS RING TOPOLOGY :

- All the nodes are connected to the single cable called bus. Every workstation communicates with the other device through this Bus.
- A signal from the source is broadcasted and it travels to all workstations connected to bus cable.
- only the intended recipient, whose MAC address or IP address matches, accepts it.
- If the MAC /IP address of machine doesn't match with the intended address, machine discards the signal.



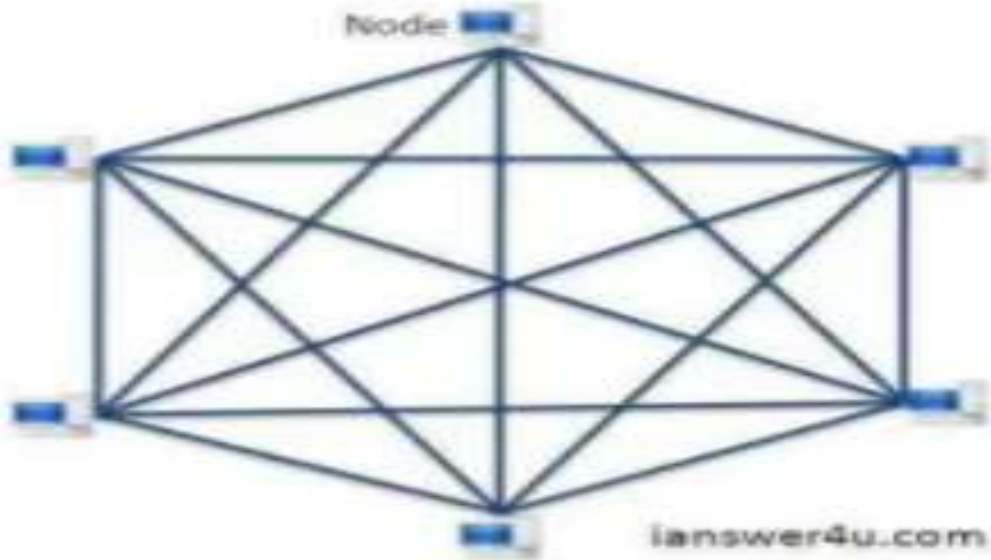
STAR TOPOLOGY :

- In Star topology, all the components of network are connected to the central device called "hub" which may be a hub, a router or a switch.
- Hub acts as a junction to connect different nodes present in Star Network, and at the same time it manages and controls whole of the network.



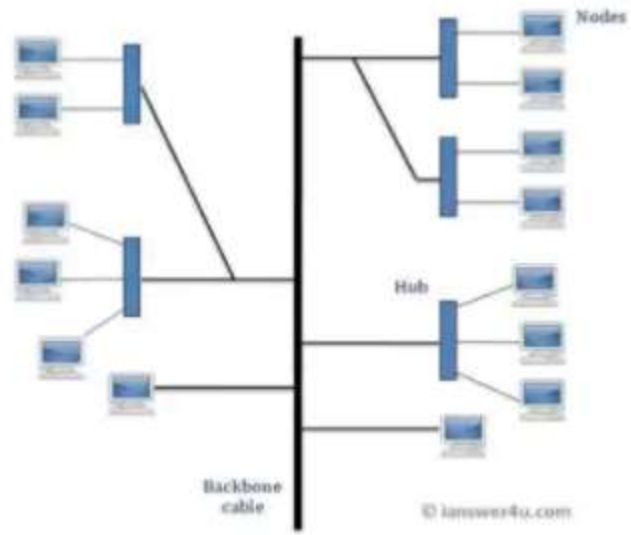
MESH TOPOLOGY :

- In a mesh network topology, each of the network node, computer and other devices, are interconnected with one another.
- Every node not only sends its own signals but also relays data from other nodes.



TREE TOPOLOGY :

- Tree Topology integrates the characteristics of Star and Bus Topology.
- number of Star networks are connected using Bus.



IEEE 802.3Ethernet

1.IEEE STANDARDS

- The relationship of the 802 Standard to the traditional OSI model is shown in Fig-
- ure 13.1.
- The IEEE has subdivided the data link layer into two sublayers:
  - logical link control (LLC)
  - media access control (MAC).
- IEEE has also created several physical layer standards for different LAN protocols.

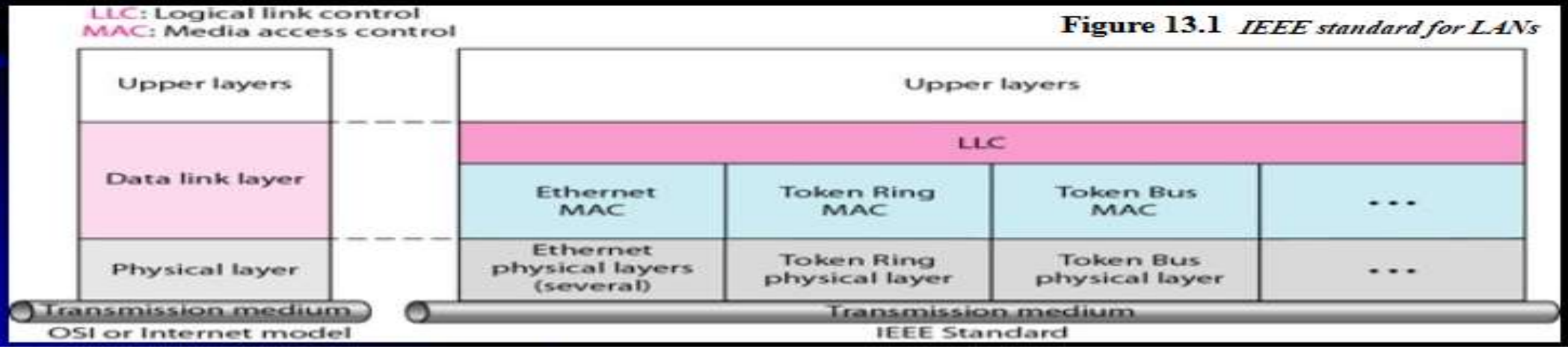


Figure 13.1 *IEEE standard for LANs*

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

logical link control (LLC)

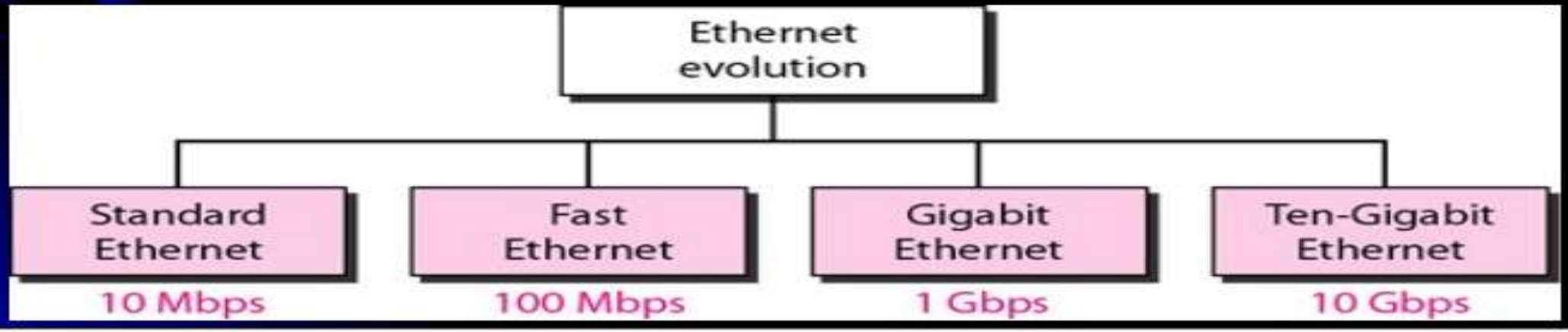
flow control, error control, and part of the framing duties are collected into one sub layer called the logical link control

Media access control (MAC)

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 defines the token-passing method for Token Ring and Token Bus LANs.

Standard Ethernet

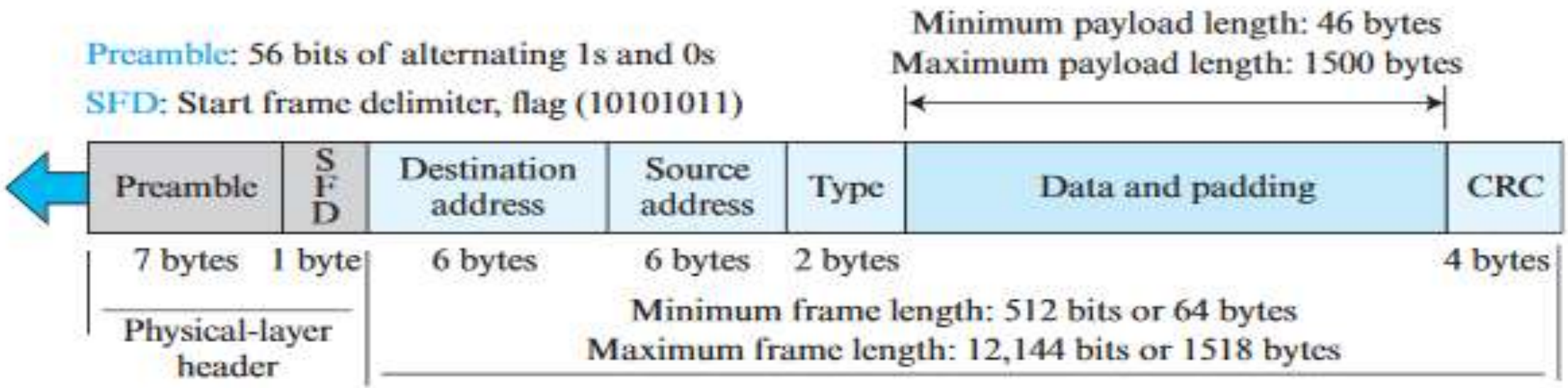
- The original Ethernet was created in 1976 at Xerox's Palo Alto Research Center (PARC).
- Since then, it has gone through four generations:
  - Standard Ethernet (10 Mbps)
  - Fast Ethernet (100 Mbps)
  - Gigabit Ethernet (1 Gbps)
  - Ten-Gigabit Ethernet (10 Gbps)



Frame Format

The Ethernet frame contains seven fields, as shown in Figure 5.55.

Figure 5.55 *Ethernet frame*



- Preamble.
  - The first field of the 802.3 frame
    - contains 7 bytes (56 bits) of alternating 0 s and 1 s
    - The pattern provides only an alert and a timing pulse.
      - 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 (1 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.
- 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.
  - 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.
  - 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 Figure 13.5.
- The minimum length restriction is required for the correct operation of CSMA/CD
  - An Ethernet frame needs to have a minimum length of 512 bits or 64 bytes.
  - Part of this length is the header and the trailer.
  - If we count 18 bytes of header and trailer
    - 6 bytes of source address,
    - 6 bytes of destination address,
    - 2 bytes of length or type,
    - 4 bytes of CRC),
  - then the minimum length of data from the upper layer is 64 - 18 = 46 bytes.
  - If the upper-layer packet is less than 46 bytes, padding is added to make up the difference.

Frame Length

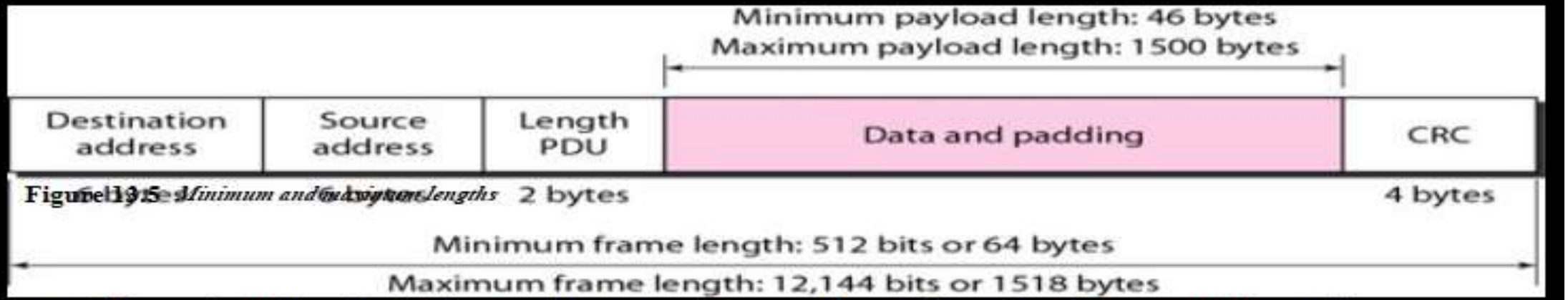


Figure 13.5 *Minimum and maximum lengths*

- The standard defines the maximum length of a framewithout) preamble and SFD (field as 1518.bytes
  - If we subtract the 18 bytes of header and ,trailer the maximum lengthof the payload is 1500.bytes
  - The maximum length restriction has two historical.reasons
  - ,First memory was very expensive when Ethernet was:designed
    - a maximum length restriction helped to reduce the size of the.buffer
  - ,Second the maximum length restrictionprevents
    - one station from monopolizing the shared,medium
    - blocking other stations that have data to.send

Minimum frame length: 64 bytes	Minimum data length: 46 bytes
Maximum frame length: 1518 bytes	Maximum data length: 1500 bytes

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.
- As shown in Figure 13.6, the Ethernet address is 6 bytes
- (48 bits), normally written in hexadecimal notation, with a colon between the bytes.

Figure 13.6 *Example of an Ethernet address in hexadecimal notation*

Addressing

- Unicast, Multicast, and Broadcast Addresses
- A source address is always a unicast address
  - the frame comes from only one station.
  - The destination address, can be unicast, multicast, or broadcast.
  - Figure 13.7 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.

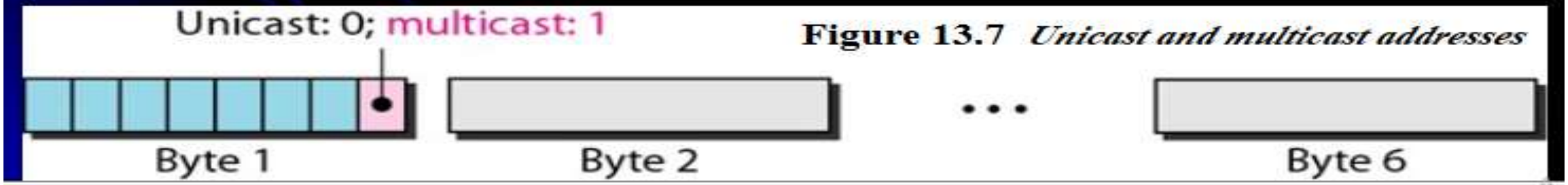


Figure 13.7 *Unicast and multicast addresses*

Addressing

- A unicast destination address defines only one recipient; the relationship between the sender and the receiver is one-to-one.
- A multicast destination address defines a group of addresses; the relationship between the sender and the receivers is one-to-many.
- The broadcast address is a special case of the multicast address; the recipients are all the stations on the LAN. A broadcast destination address is forty-eight 1s.

Example 13.1

- Define the type of the following destination addresses:
  a. 4A:30:10:21:10:IA
  b. 47:20:1B:2E:08:EE
  c. FF:FF:FF:FF:FF:FF

Unicast, Multicast, and Broadcast Addresses

Solution

- we need to look at the second hexadecimal digit from the left.
  - If it is even, the address is unicast.
  - If it is odd, the address is multicast.
  - If all digits are F's, the address is broadcast.
- Therefore, we have the following:

  a. This is a unicast address because A in binary is 1010 (even).

  b. This is a multicast address because 7 in binary is 0111 (odd).

  c. This is a broadcast address because all digits are F's.
- The way the addresses are sent out on line is different from the way they are written in hexadecimal notation. The transmission is left-to-right, byte by byte; however, 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.

Unicast, Multicast, and Broadcast Addresses

Example 13.2

- Show how the address 47:20:IB:2E:08:EE is sent out on line.

Solution

- The address is sent left-to-right, byte by byte; for each byte, it is sent right-to-left, bit by bit, as

shown below:

11100010 00000100 11011000 01110100 00010000 01110111

The least significant bit of the first byte defines the type of address.
If the bit is 0, the address is unicast; otherwise, it is multicast.

The broadcast destination address is a special case of the multicast address in which all bits are 1s.

Access Method: CSMA/CD

- Standard Ethernet uses 1-persistent CSMA/CD (see Chapter 12).
- Slot Time
  - 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.
  - Slot time = round-trip time + time required to send the jam sequence
- 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 51.2 μs.
- Slot Time and Collision
  - The choice of a 512-bit slot time was not accidental.
  - It was chosen to allow the proper functioning of CSMA/CD.
  - To understand the situation, let us consider two cases.
  - In the first case, we assume that the sender sends a minimum-size packet of 512 bits.
    - Before the sender can send the entire packet out, the signal travels through the network and reaches the end of the network.
    - If there is another signal at the end of the network (worst case), a collision occurs.

Access Method: CSMA/CD

- The sender has the opportunity to abort the sending of the frame and to send a jam sequence to inform other stations of the collision.
- The round-trip time plus the time required to send the jam sequence should be less than the time needed for the sender to send the minimum frame, 512 bits.
- The sender needs to be aware of the collision before it is too late, that is, before it has sent the entire frame.

- In the second case, the sender sends a frame larger than the minimum size (between 512 and 1518 bits).
  - In this case, if the station has sent out the first 512 bits and has not heard a collision, it is guaranteed that collision will never occur during the transmission of this frame.

The reason is that the signal will reach the end of the network in less than one-half the slot time.

- If all stations follow the CSMA/CD protocol, they have already sensed the existence of the signal (carrier) on the line and have refrained from sending.
- If they sent a signal on the line before one-half of the slot time expired,
  - a collision has occurred and the sender has sensed the collision.
  - In other words, collision can only occur during the first half of the slot time, and if it does, it can be sensed by the sender during the slot time.
  - This means that after the sender sends the first 512 bits, it is guaranteed that collision will not occur during the transmission of this frame.
  - The medium belongs to the sender, and no other station will use it.
  - In other words, the sender needs to listen for a collision only during the time the first 512 bits are sent.

Access Method: CSMA/CD

- Of course, all these assumptions are invalid if a station does not follow the CSMA/CD protocol.
- In this case, we do not have a collision, we have a corrupted station.

Slot Time and Maximum Network Length

- There is a relationship between the slot time and the maximum length of the network (collision domain).
- It is dependent on the propagation speed of the signal in the particular medium.
- In most transmission media, the signal propagates at 2 x 10^8 m/s (two-thirds of the rate for propagation in air).
- For traditional Ethernet, we calculate
  - MaxLength = PropagationSpeedx SlotTime / 2
  - MaXLength = (2 X 10^8) X(51.2 x 10^{-6}/2(= 5120 m
- We need to consider
  - the delay times in repeaters and interfaces,
  - and the time required to send the jam sequence.
  - These reduce the maximum-length of a traditional Ethernet network to 2500 m, just 48 percent of the theoretical calculation.

Implementation of standard Ethernet

Implementation

The Standard Ethernet defined several implementations, but only four of them became popular during the 1980s. Table 5.6 shows a summary of Standard Ethernet implementations.

Table 5.6 *Summary of Standard Ethernet implementations*

Implementation	*Medium*	*Medium Length*	*Encoding*
10Base5	Thick coax	500 m	Manchester
10Base2	Thin coax	185 m	Manchester
10Base-T	2 UTP	100 m	Manchester
10Base-F	2 Fiber	2000	Manchester

Fast Ethernet (100 Mbps)

- IEEE created Fast Ethernet (802.3u)
- Fast Ethernet is backward-compatible with Standard Ethernet,
- it transmit data at a rate of 100 Mbps.
- The goals of Fast Ethernet can be summarized as follows:
    1. Upgrade the data rate to 100 Mbps.
    2. Make it compatible with Standard Ethernet.
    3. Keep the same 48-bit address.
    4. Keep the same frame format.
    5. Keep the same minimum and maximum frame lengths.

MAC Sublayer

- It uses the MAC sublayer untouched.
- It uses the star topology with half duplex and full duplex.
- In the half-duplex approach,
  - the stations are connected via a hub;
- in the full-duplex approach,
  - the connection is made via a switch with buffers at each port.
- It uses the CSMA/CD for the half-duplex approach;
- for full-duplex Fast Ethernet, there is no need for CSMA/CD.
- However, the implementations keep CSMA/CD for backward compatibility with Standard Ethernet.

Autonegotiation

- It allows a station or a hub a range of capabilities.
- Autonegotiation allows two devices to negotiate the mode or data rate of operation.
- It was designed particularly for the following purposes:
  - To allow incompatible devices to connect to one another.
    - 10 Mbps with 100 Mbps
  - To allow one device to have multiple capabilities.
  - To allow a station to check a hub's capabilities.

Characteristics	*100Base-TX*	*100Base-FX*	*100Base-T4*
Media	Cat 5 UTP or STP	Fiber	Cat 4 UTP
Number of wires	2	2	4
Maximum length	100 m	100 m	100 m
Block encoding	4B/5B	4B/5B	
Line encoding	MLT-3	NRZ-I	8B/6T

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.
- The goals of the Gigabit Ethernet design can be summarized as follows:
  1. Upgrade the data rate to 1 Gbps.
  2. Make it compatible with Standard or Fast Ethernet.
  3. Use the same 48-bit address.
  4. Use the same frame format.
  5. Keep the same minimum and maximum frame lengths.
  6. To support autonegotiation as defined in Fast Ethernet.

Table 13.3 *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

GIGABIT ETHERNET

Ten-Gigabit Ethernet

- The IEEE committee created Ten-Gigabit Ethernet and called it Standard 802.3ae.
- The goals of the Ten-Gigabit Ethernet design can be summarized as follows:
  1. Upgrade the data rate to 10 Gbps.
  2. Make it compatible with Standard, Fast, and Gigabit Ethernet.
  3. Use the same 48-bit address.
  4. Use the same frame format.
  5. Keep the same minimum and maximum frame lengths.
  6. Allow the interconnection of existing LANs into a metropolitan area network (MAN) or a wide area network (WAN).
  7. Make Ethernet compatible with technologies such as Frame Relay and ATM

GIGABIT ETHERNET

MAC Sublayer

- Ten-Gigabit Ethernet operates only in full duplex mode which means there is no need for contention;
- CSMA/CD is not used in Ten-Gigabit Ethernet.

Physical Layer

- The physical layer in Ten-Gigabit Ethernet is designed for using fiber-optic cable over long
- distances.
- Three implementations are the most common:
  - 10GBase-S,
  - 10GBase-L,
  - and 10GBase-E.

Table 13.4 *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