



THE UNIVERSITY OF  
MELBOURNE

# Week6 Data-Link Layer and Network Layer

COMP90007 Internet Technology

---

Prepared by: Chenyang Lu (Luke)





# Your Tutor

## Chenyang Lu (Luke)

- Email: [chenyang.lu@unimelb.edu.au](mailto:chenyang.lu@unimelb.edu.au)
- Workshop Slides: <https://github.com/LuChenyang3842/Internet-technology-teaching-material>

Day	Time	Location
Tue	18:15	Bouverie st –B114
Wed	10:00	Elec Engineering -122
Wed	17:15	Bouverie-sr 132



# Data-Link Layer

# Data Link Layer



## Function of Data link layer:

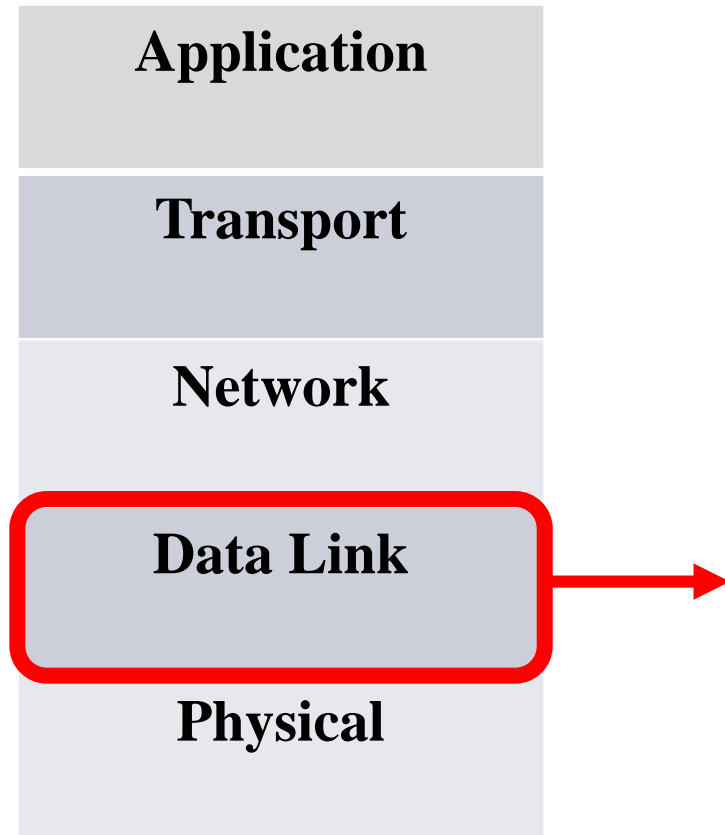
- Provide service to network layer
- Transmission control
- Error Control

## Primary Method:

- Take packets from network layer
- Encapsulate them into frames

This terminology only used in Datalink Layer

# Data Link Layer



1. Framing methods
  - Character Count
  - Flag Bytes with Byte stuffing
  - Flag with Bit stuffing
2. Error Control
  - Error Bounds (Hamming Distance)
  - Detecting (parity, checksum, CRC )
  - Correcting (Hamming code)
  - Re-transmission
3. Flow Control
  - Feedback Based Flow Control
    - Stop and wait
    - Sliding window
  - ~~Rate based flow control~~



# Error Control – Error Bound (Hamming distance)

## Hamming distance :

- minimum bit flips to turn one valid codeword into another valid one
- Correct d errors :  $d = \frac{n-1}{2}$  (n: hamming distance)
- Detect d error where  $d = n - 1$  (n: hamming distance)

# Error Control – Detecting (parity)

	0	0	1	1
$\wedge$	0	1	0	1
	0	1	1	0

## □ Parity(1bit)

- **Even parity (two different approaches)**

- Approach1: XOR every bit (including parity) to be 0
- Approach2: For a given set of bits, The occurrences of bits whose value is 1 is counted. If that count is odd, the parity bit value is set to 1, making the total count of occurrences of 1s in the whole set (including the parity bit) an even number. If the count of 1s in a given set of bits is already even, the parity bit's value is 0.

- **Odd Parity (two different approaches)**

- Approach1: XOR every bit (including parity) to be 1
- Approach2: For a given set of bits, if the count of bits with a value of 1 is even, the parity bit value is set to 1 making the total count of 1s in the whole set (including the parity bit) an odd number. If the count of bits with a value of 1 is odd, the count is already odd so the parity bit's value is 0.

**Note:** For both even parity and odd parity, I have provided two different approaches and they will have the same result, you can pick either of them which is easier for you to understand.



# Error Control – Detecting (parity)

## □ Parity(1bit)

- Hamming distance = 2
- Example for even parity:

11111110

11111101



# Error Control – Error Detecting Code (Cyclic Redundancy check)

## ❑ Cyclic Redundancy Check

- Based on **generator polynomial  $G(x)$**  --- (Don't need to know the detail of  $G(x)$ , will be provided)
  - Determine **bits polynomial** based on generator,
    - For  $G(x) = x^4 + x^1 + 1$ , the bits polynomial is 10011.
    - For  $G(x) = x^5 + x^2 + 1$ , the bits polynomial is 100101
  - Determine **r**. (The degree of  $G(x)$ , same as the length of bits polynomial -1 )
    - For  $G(x) = x^4 + x^1 + 1$ ,  $r = 4$
    - For  $G(x) = x^5 + x^2 + 1$ ,  $r = 5$
  - Append  $r$  zeros into the frame
  - using modulo 2 division** (frame with  $r$  zero appended divided by bits polynomial)
  - subtract the  $r$  bits remainder Frame with  $r$  zero appended using **modulo 2 subtraction**.  
(Same as append remainder  $r$  bits to the end of frame)
- Hamming distance = 4



## Question2 - CRC

Using the polynomial code method, compute the CRC for the frame: 1101011111 having a generator polynomial  $G(x) = x^4 + x^1 + 1$

Step1: generate bits polynomial: 10011

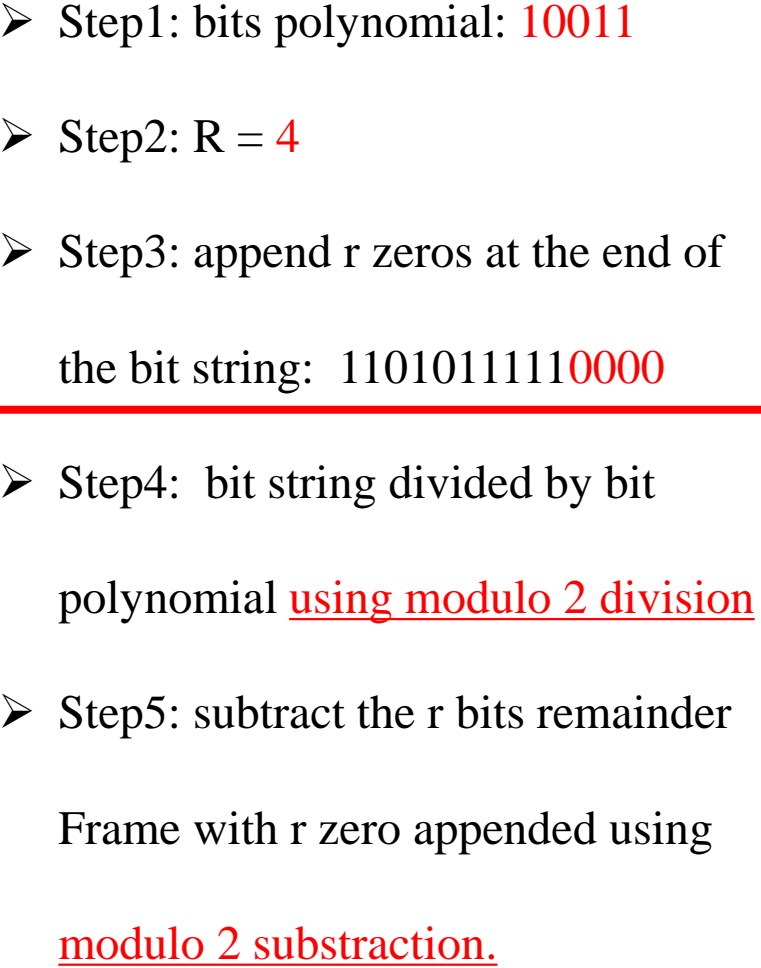
Step2: determine  $r = 4$

Step3: append  $r$  zeros at the end of the bit string: 11010111110000

Step4: bit string divided by bit polynomial using modulo 2 division

Step5: subtract the  $r$  bits remainder Frame with  $r$  zero appended using modulo 2 subtraction.

(Same as append remainder  $r$  bits to the end of frame)



On the receiver side, they use transmitted data divided by bit polynomial using modulo 2 division

- if the remainder is 0, there no error detected
- If the remainder is not 0, some error occurs during transmission

Video of CRC (12minutes): <https://www.youtube.com/watch?v=6gbkoFciryA>

Modulo 2 Division and subtraction: <https://hubpages.com/technology/Modulo-2-Arithmetic>

# Error Control – Error Correcting Code

## Hamming code

- Hamming distance = 3
- We need to put check bits in position p that are power of 2
- Use even parity to determining check bits.

A byte of data: 10011010

Create the data word, leaving spaces (at the position which is power of 2) for the parity bits: \_ \_ 1 \_ 0 0 1 \_ 1 0 1 0

- Position 1 checks bits 1,3,5,7,9,11

? \_ 1 \_ 0 0 1 \_ 1 0 1 0. so set position 1 to 0. 0 \_ 1 \_ 0 0 1 \_ 1 0 1 0.

- Position 2 checks bits 2,3,6,7,10,11

0 ? 1 \_ 0 0 1 \_ 1 0 1 0. so set position 2 to 1. 0 1 1 \_ 0 0 1 \_ 1 0 1 0.

- Position 4 checks bits 4,5,6,7,12:

0 1 1 ? 0 0 1 \_ 1 0 1 0. so set position 4 to 1: 0 1 1 1 0 0 1 \_ 1 0 1 0.

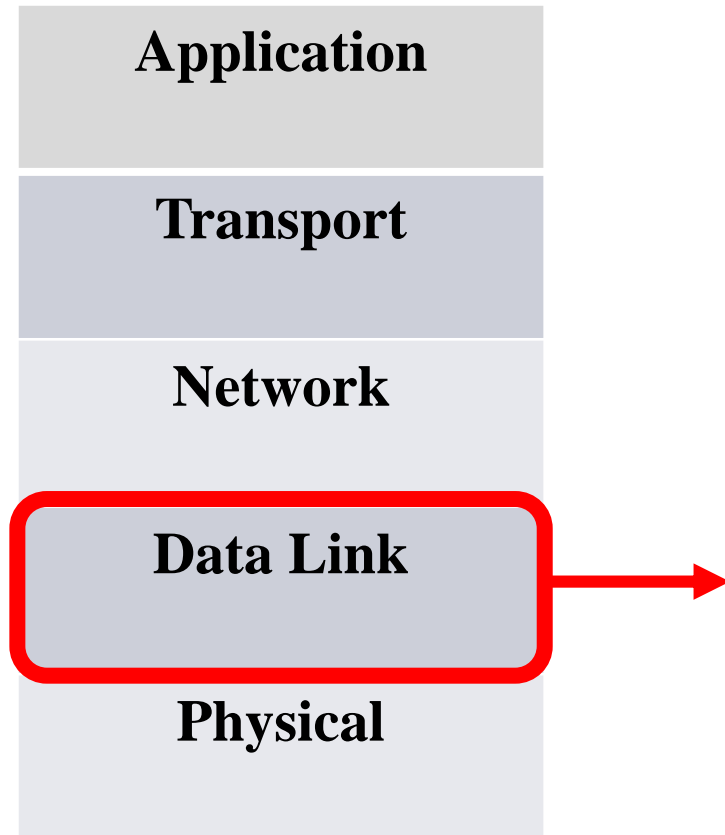
- Position 8 checks bits 8,9,10,11,12:

0 1 1 1 0 0 1 ? 1 0 1 0. so set position 8 to 0: 0 1 1 1 0 0 1 0 1 0 1 0.

Codeword: 011100101010

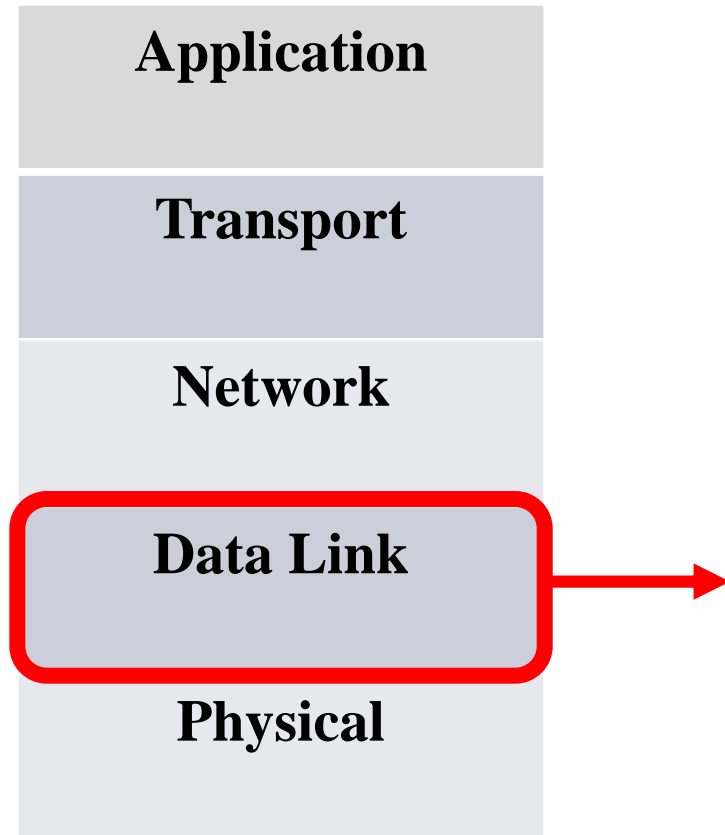
# Data Link Layer

## TCP/IP Model



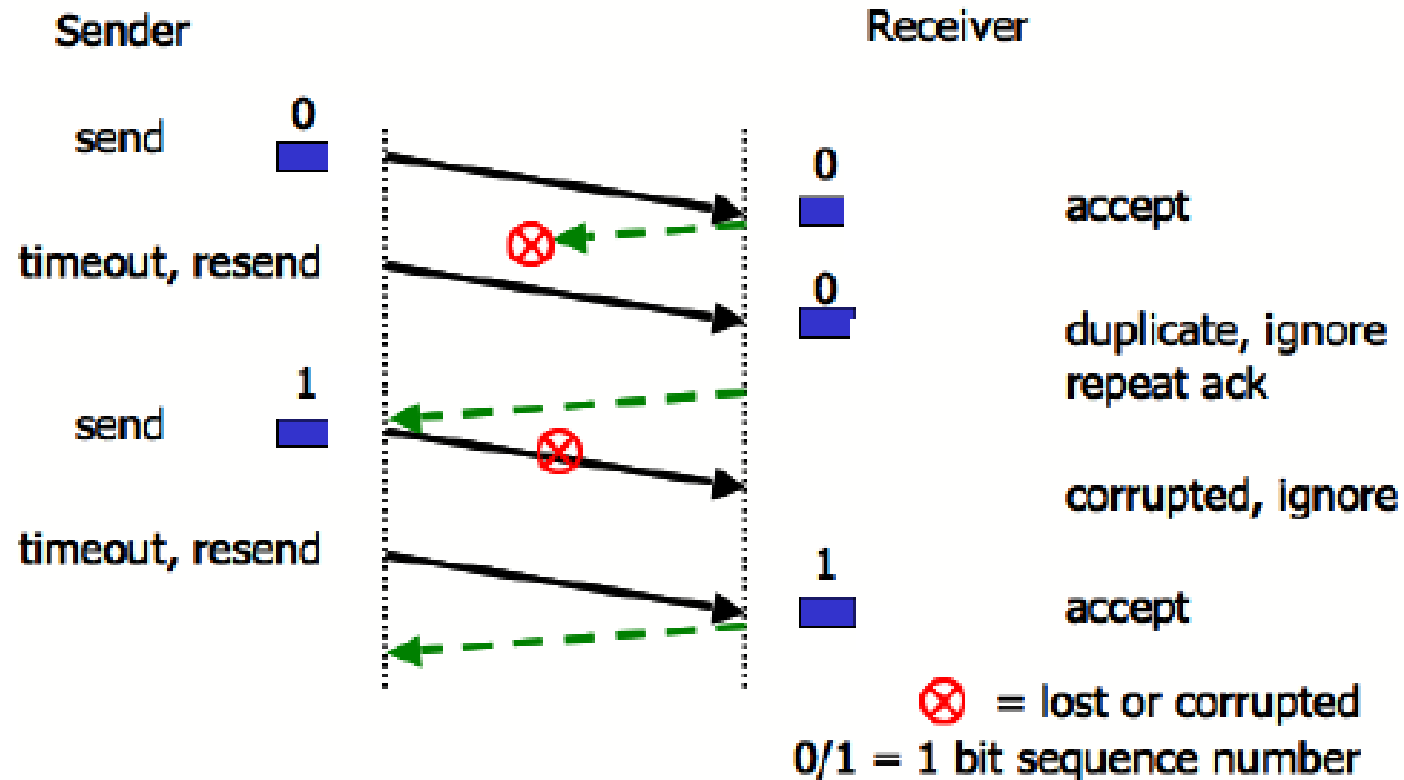
1. Framing methods
  - Character Count
  - Flag Bytes with Byte stuffing
  - Flag with Bit stuffing
2. Error Control
  - Error Bounds (Hamming Distance)
  - Detecting (parity, checksum, CRC )
  - Correcting (Hamming code)
  - Re-transmission
3. Flow Control
  - Feedback Based Flow Control
    - Stop and wait
    - Sliding window
  - ~~Rate based flow control~~

# Data Link Layer



1. Framing methods
  - Character Count
  - Flag Bytes with Byte stuffing
  - Flag with Bit stuffing
2. Error Control
  - Error Bounds (Hamming Distance)
  - Detecting (parity, checksum, CRC )
  - Correcting (Hamming code)
  - Re-transmission
3. Flow Control
  - Feedback Based Flow Control
    - **Stop and wait**
    - Sliding window
  - ~~Rate based flow control~~

# Flow Control – Stop and wait



- Send one data unit at one time
- If Ack, then send another.
- If timeout, send the same one again



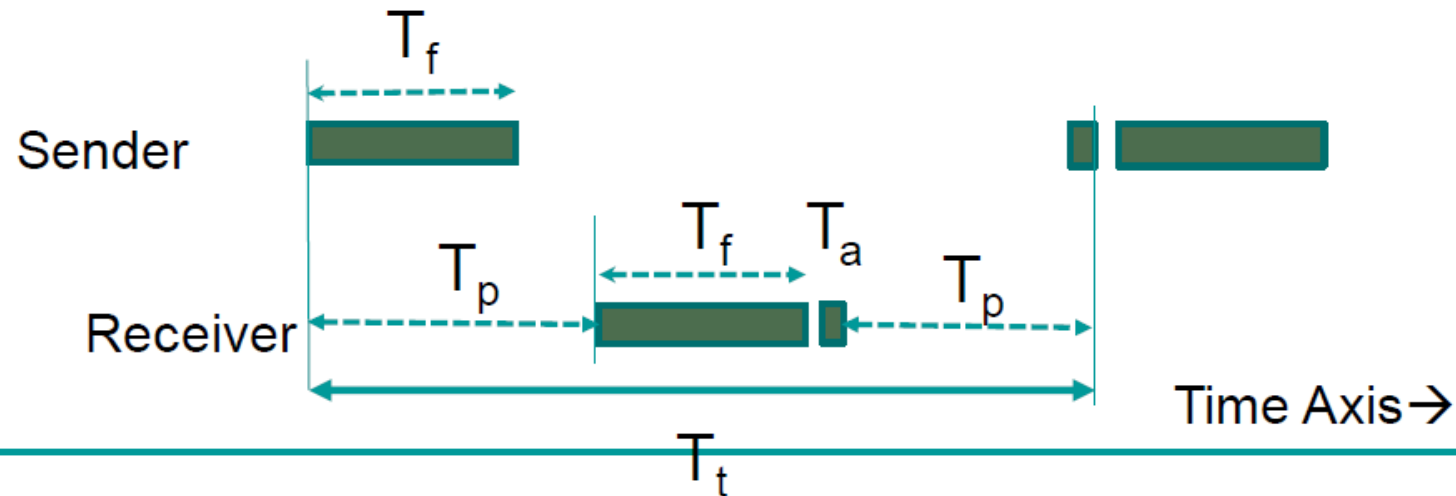
# Flow Control – Stop and wait (Efficiency)

$$\text{Efficiency} = \frac{\text{Transmission Delay}}{\text{Total Latency}} = \frac{T_f}{T_t}$$

$$T_f = \frac{\text{Length(size) of Frame}}{\text{Bit rate}} = \frac{L}{B}$$

$$T_t = T_f + 2T_p + T_a = T_f + 2T_p$$

$T_a$  = Time for transmitting an ACK, assume to be zero



# Question1

A channel has a bit rate of 4 kbps and a propagation delay of 20 ms. For what range of frame sizes does stop-and-wait give an efficiency of at least 50 percent?

$$\text{Efficiency} = \frac{\text{Transmission Delay}}{\text{Total Latency}} = \frac{T_f}{T_t}$$

$$T_f = \frac{\text{Length(size) of Frame}}{\text{Bit rate}} = \frac{L}{B}$$

$$T_t = T_f + 2T_p$$

# Question1

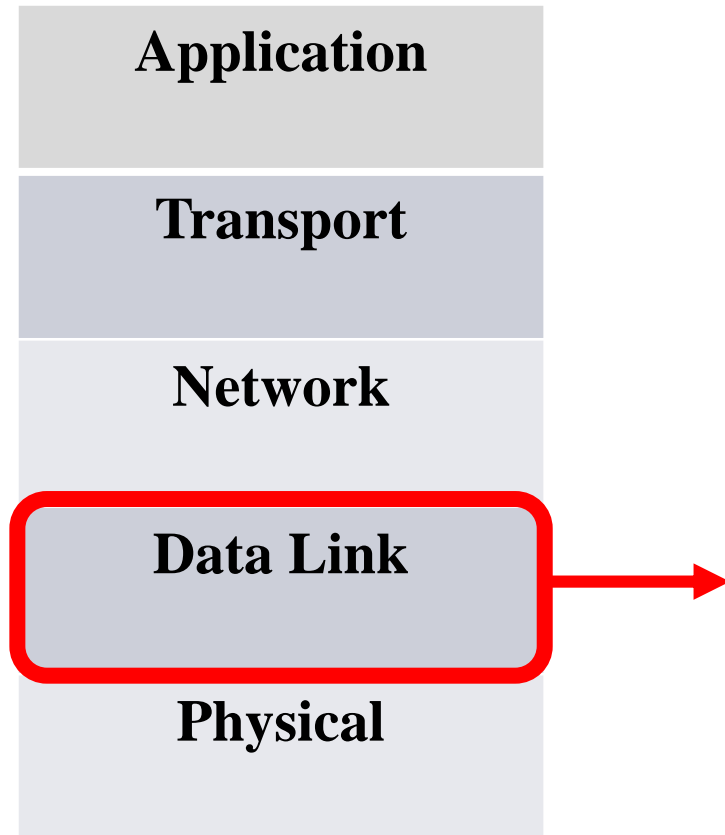
A channel has a bit rate of 4 kbps and a propagation delay of 20 ms. For what range of frame sizes does stop-and-wait give an efficiency of at least 50 percent?

$$Efficiency = \frac{Tranmission\ Delay}{Total\ Latency} = \frac{T_f}{T_t}$$

$L \geq 160\text{bits}$

The frame size should be equal to or larger than 160 bits

# Data Link Layer

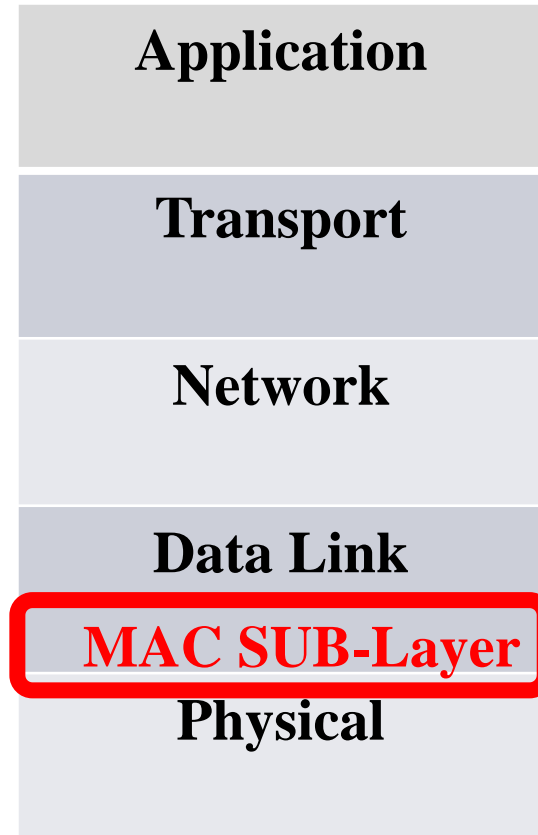


1. Framing methods
  - Character Count
  - Flag Bytes with Byte stuffing
  - Flag with Bit stuffing
2. Error Control
  - Error Bounds (Hamming Distance)
  - Detecting (parity, checksum, CRC )
  - Correcting (Hamming code)
  - Re-transmission
3. Flow Control
  - Feedback Based Flow Control
    - Stop and wait
    - Sliding window
  - ~~Rate based flow control~~



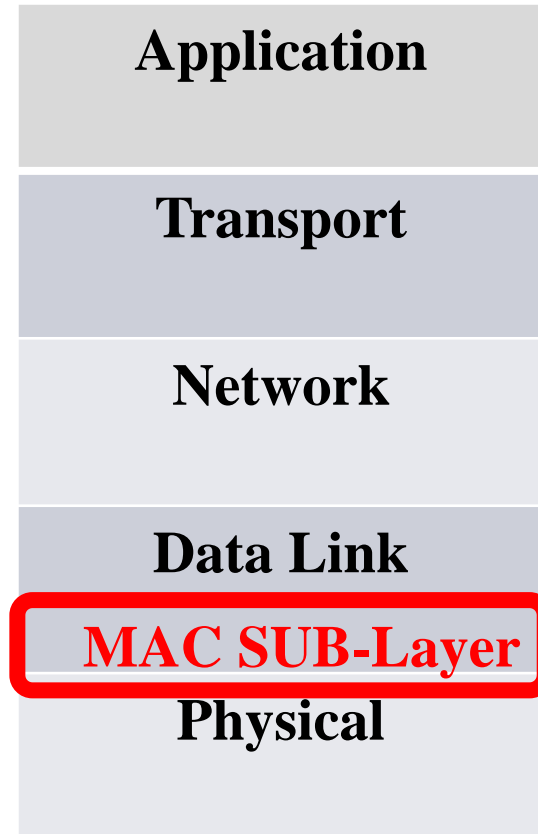
# MAC Sub-Layer

# Mac Sub-Layer



*Question: What is MAC Sub-Layer and what is the function of MAC Sub-Layer*

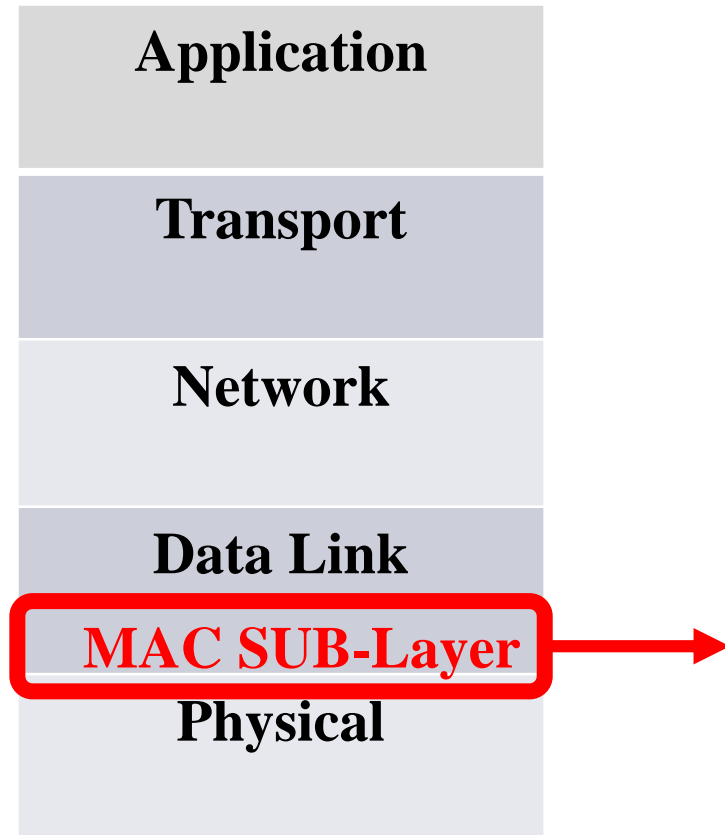
# Mac Sub-Layer



## Medium Access Control Sub-Layer:

- Lives near the bottom of data link layer
- Control how we can allocate multiple users over a single shared channel in a broadcast

# Mac Sub-Layer



1. **Contention**
  - **ALOHA**
  - **Carrier Sense Multiple Access (CSMA)**
2. Collision Free
  - CSMA/CD – Binary Countdown
  - CSMA/CD - bit map
3. Limited Contention
  - CSMA/CD - Adaptive Tree Walk Protocol
4. MACA/MACAW (for Wireless LANs)



# Contention - ALOHA

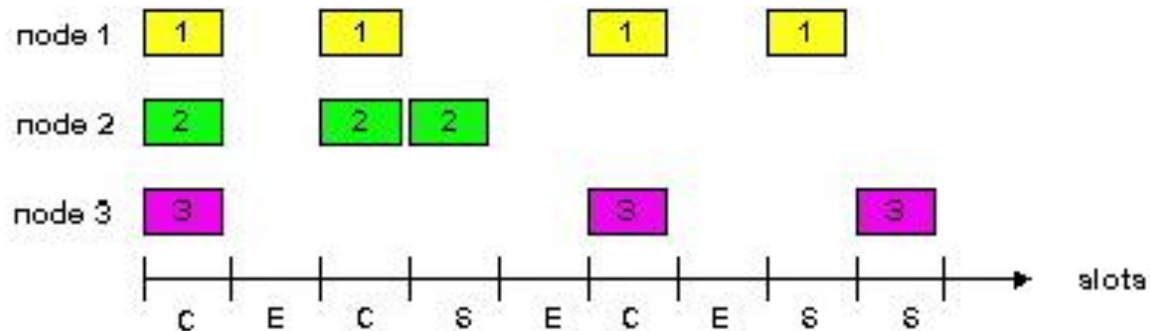
➤ Two types of ALOHA, Pure and slotted

➤ **The basic idea :**

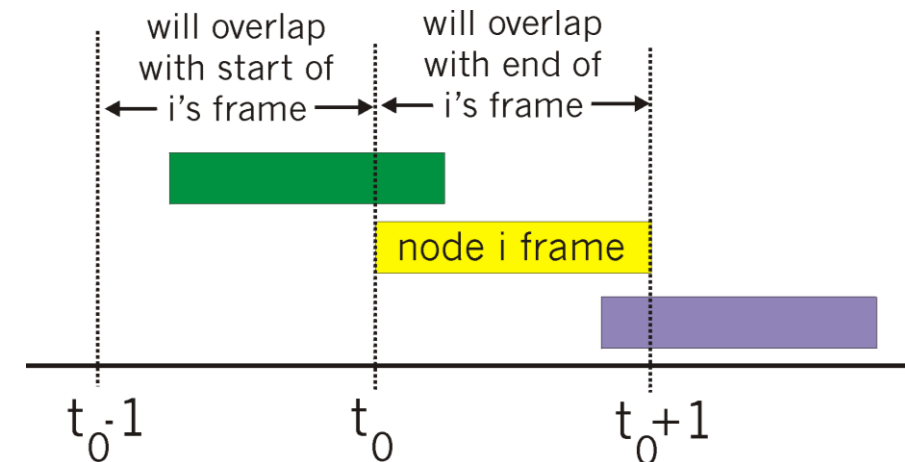
❑ Users transmit whenever they have data to be sent.

❑ Deal with collisions when they come and wait random time  
(random) and retransmit

## Slotted ALOHA

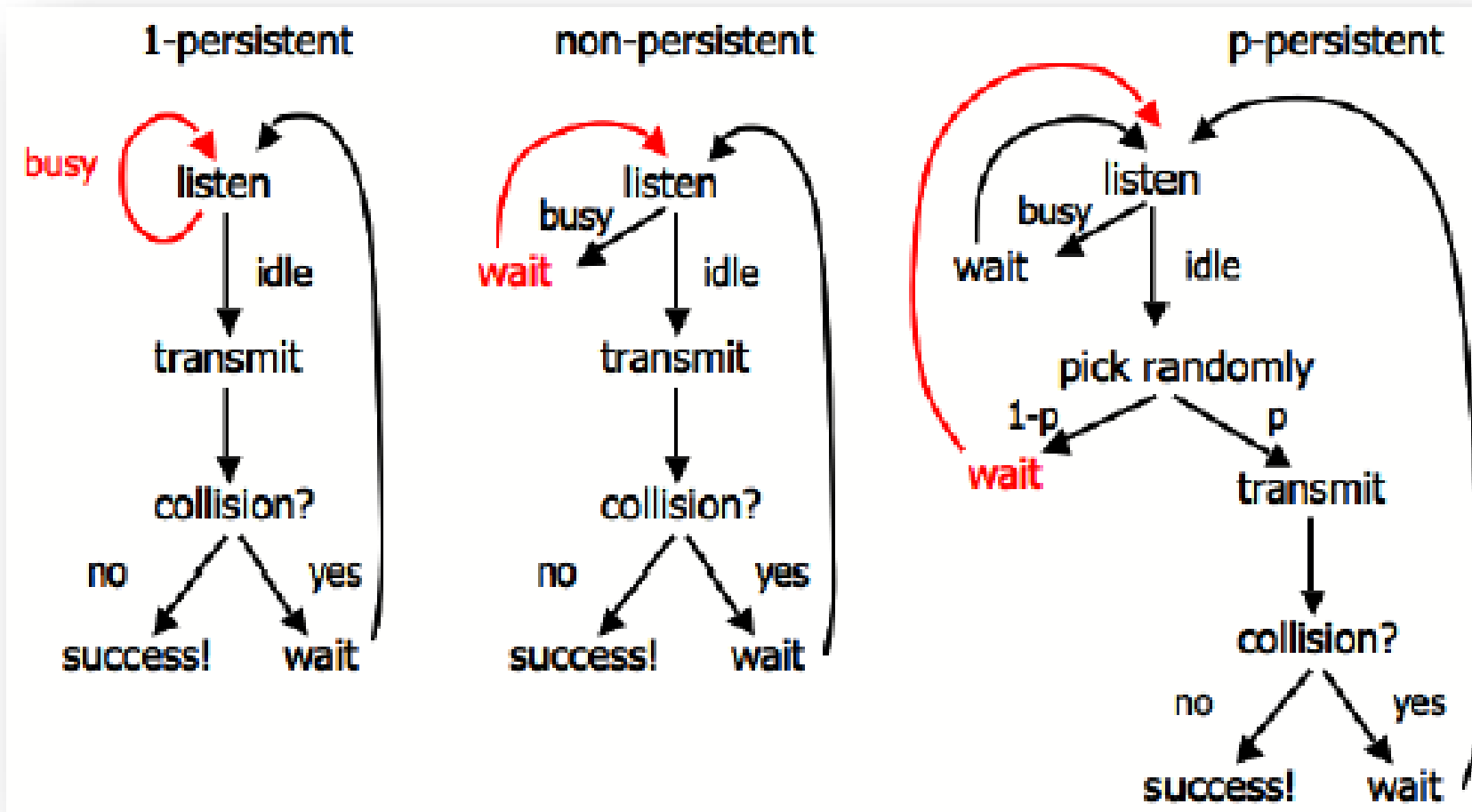


## Unslotted ALOHA



# Contention – CSMA (Carrier Sense Multiple Access)

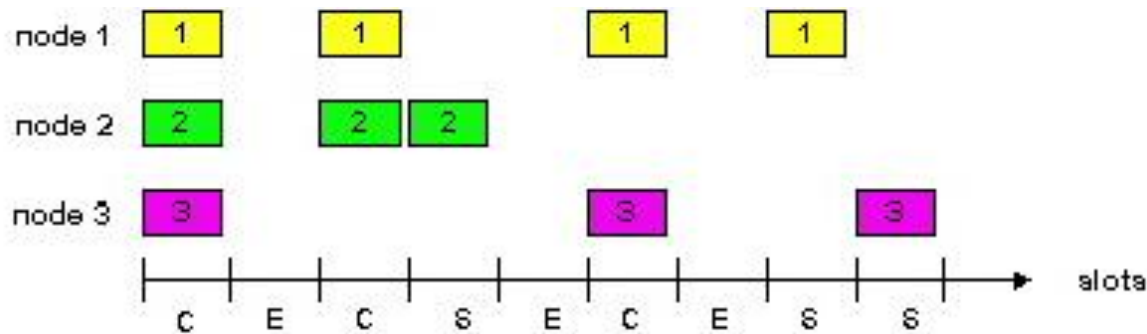
Stations listen for a carrier (i.e., a transmission) and act accordingly are called **carrier sense protocols**.



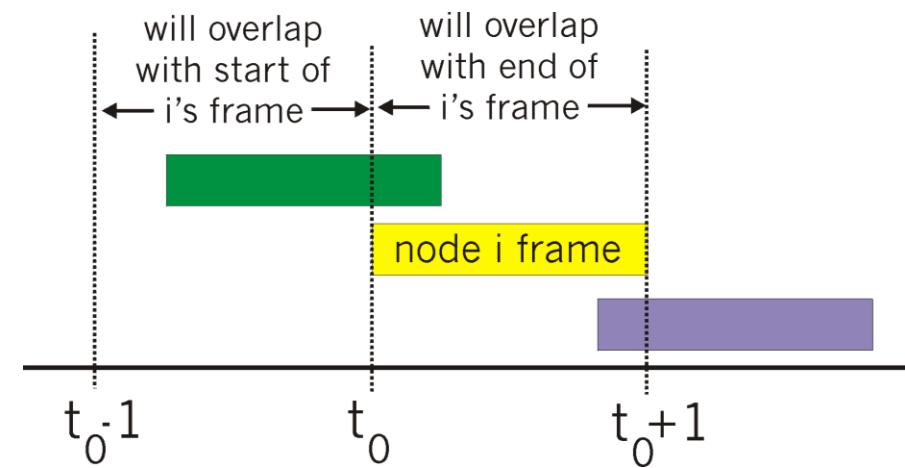
# Question3

Consider the delay of pure ALOHA versus slotted ALOHA at low load.  
Which one is less? Explain your answer.

## Slotted ALOHA



## Unslotted ALOHA



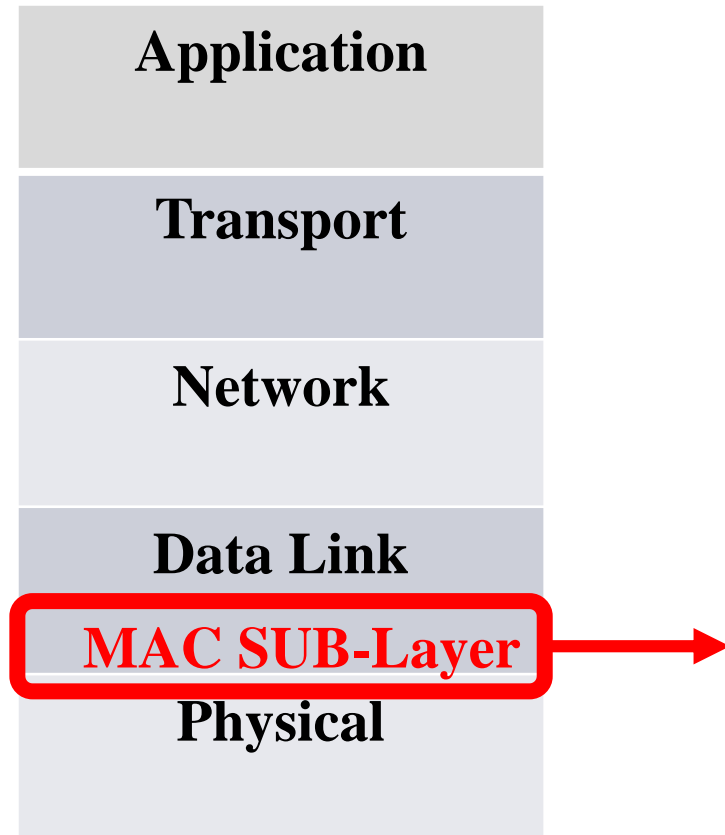
## Question3

**Consider the delay of pure ALOHA versus slotted ALOHA at low load. Which one is less? Explain your answer.**

- With slotted ALOHA, it has to wait for the next slot. This introduces half a slot time of delay.
- With pure ALOHA, transmission can start instantly. At low load with minimal collisions, pure ALOHA will have less delay.

At higher loads, there is more probability for collisions in pure ALOHA compared to slotted ALOHA. This is because frames can collide in midway. By enforcing synchronization, slotted ALOHA is able to achieve much greater efficiency.

# Mac Sub-Layer



1. Contention
  - ALOHA
  - Carrier Sense Multiple Access (CSMA)
2. **Collision Free**
  - **CSMA/CD – Binary Countdown**
  - **CSMA/CD - bit map**
3. Limited Contention
  - CSMA/CD - Adaptive Tree Walk Protocol
4. MACA/MACAW (for Wireless LANs)

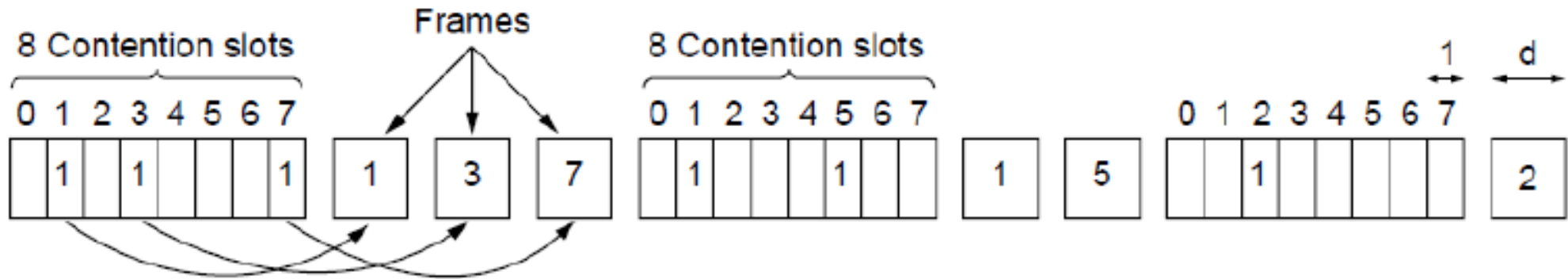
# Collision Free – CSMA with Collision Detection

- Imagine two stations sense the channel to be idle and begin transmitting simultaneously, their signals will still collide.
- Another improvement is for the stations to quickly detect the collision and abruptly stop transmitting. This protocol is known as **CSMA/CD (Collision Detection)**

# Collision Free – CSMA/CD (bit map protocol)

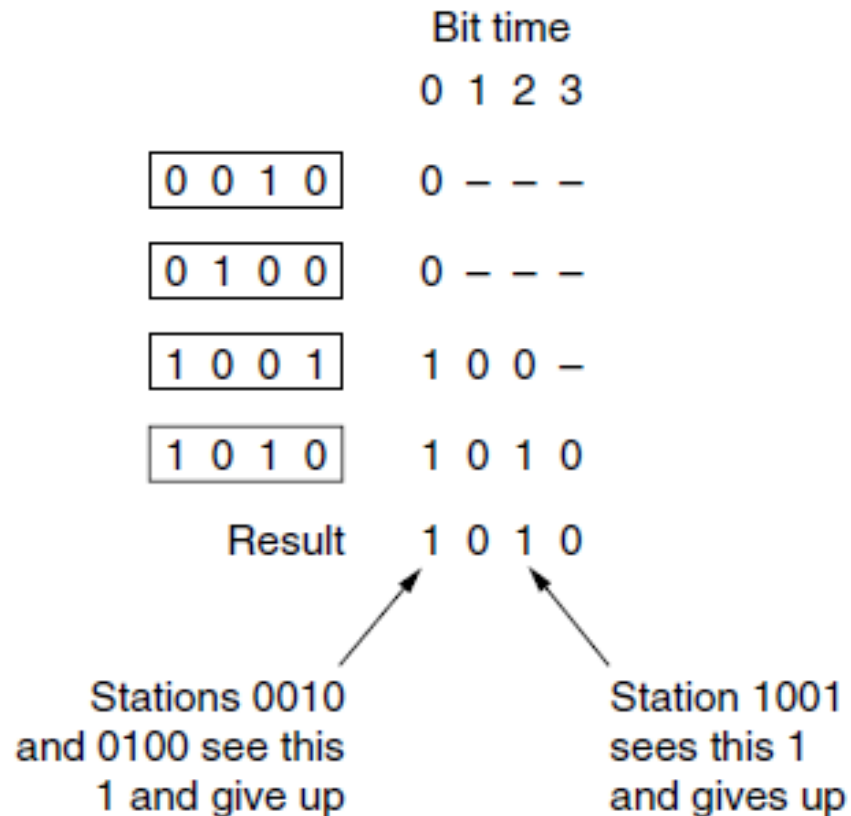
## a. Bit-map method – ‘reservation during the contention time (reservation interval)

Protocols like this in which the desire to transmit is broadcast before the actual transmission are called **reservation protocols** because they reserve channel ownership in advance



# Collision Free – CSMA/CD (Binary Countdown)

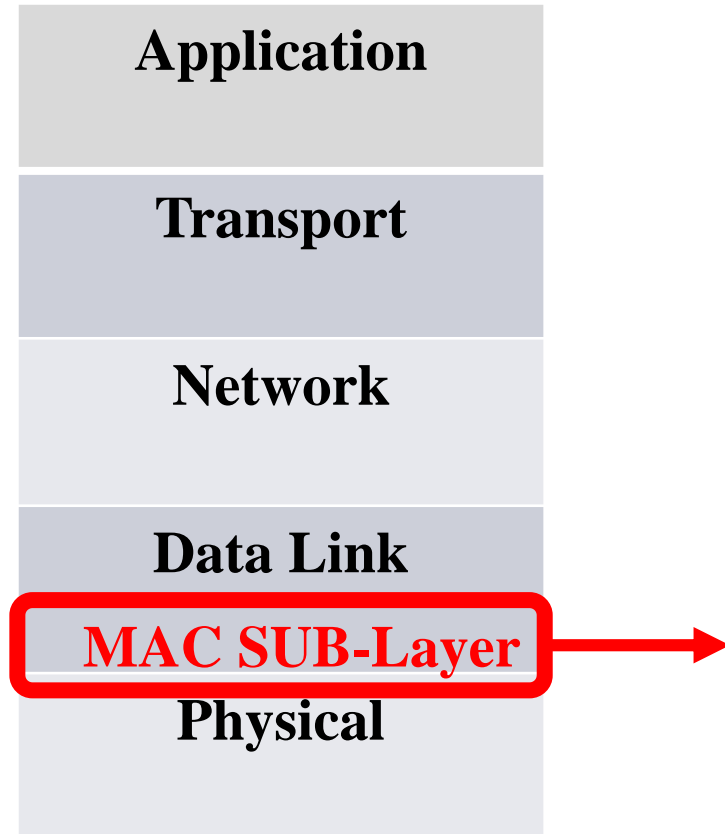
## b. Binary Count Down method



Station send their address  
in contention slot ( $\log_2 N$   
bits instead of  $N$ )



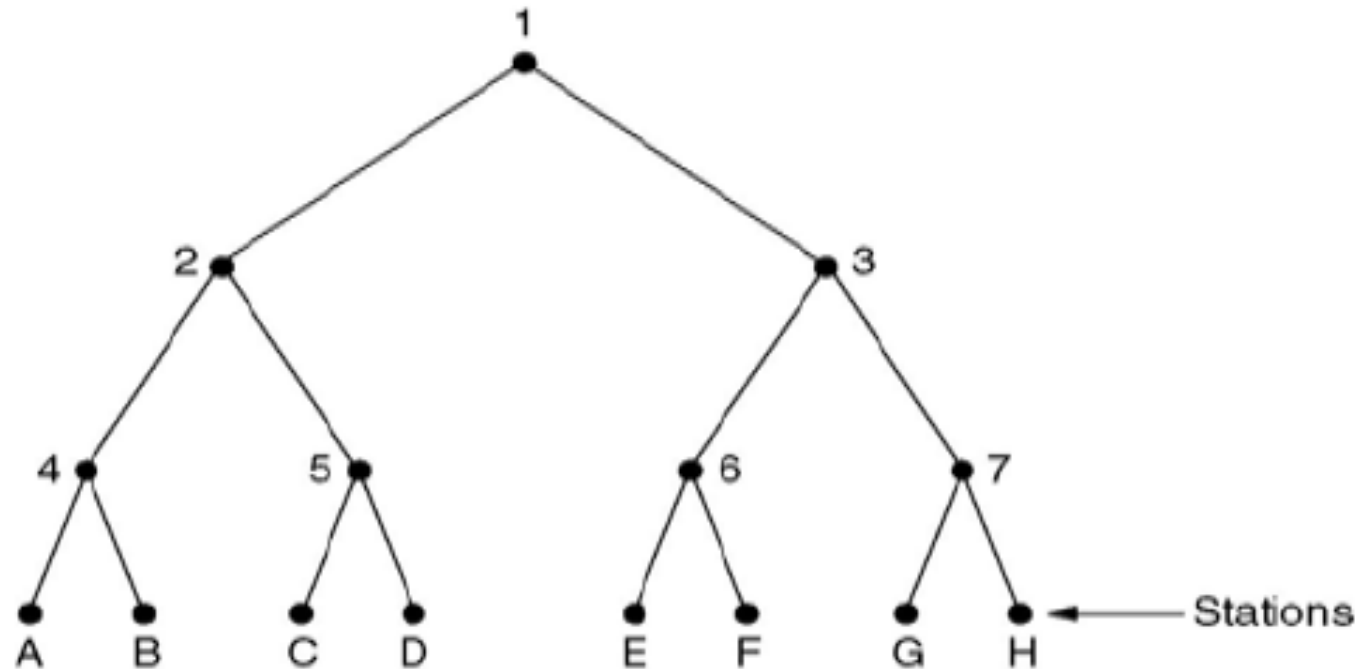
# Mac-Sub Layer



1. Contention
  - ALOHA
  - Carrier Sense Multiple Access (CSMA)
2. Collision Free
  - CSMA/CD – Binary Countdown
  - CSMA/CD - bit map
- 3. Limited Contention**
  - **CSMA/CD - Adaptive Tree Walk Protocol**
4. MACA/MACAW (for Wireless LANs)

# Limited- Contention protocols – Adaptive Tree Walk Protocol

Adaptive Tree Walk Protocol –evenly distribute the resource (depth first search)



Example 1: D G

Slot 1 → D, G – collision

Slot 2 → D

Slot 3 → G

Example 2: B D G

Slot 1 → B, D, G – collision

Slot 2 → B, D - collision

Slot 3 → B

Slot 4 → D

Slot 5 → G

## Question4

Eight stations, numbered 1 through 8, are contending for the use of a shared channel by using the adaptive tree walk protocol. If all the stations whose addresses are prime numbers suddenly became ready at once, how many slots are needed to resolve the contention?

*Answer:*

Stations 2,3,5,7 want to send. 7 slots are needed, with the contents of each slot being as follows:

slot 1: 2, 3, 5, 7 (collision)

slot 2: 2, 3 (collision)

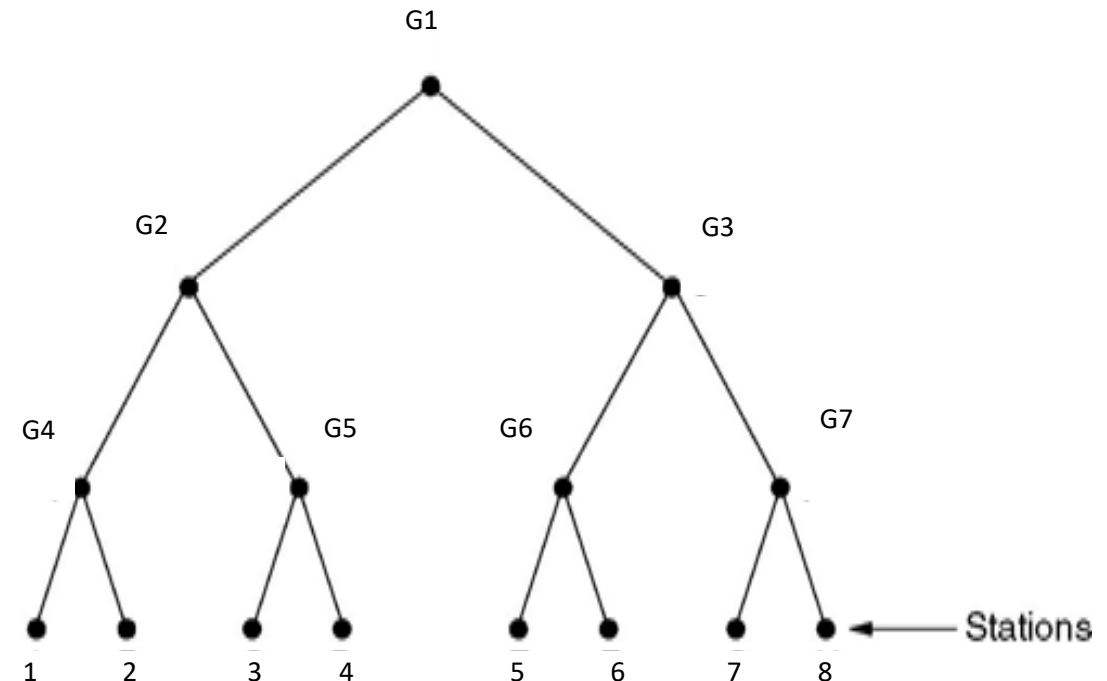
slot 3: 2 (success)

slot 4: 3 (success)

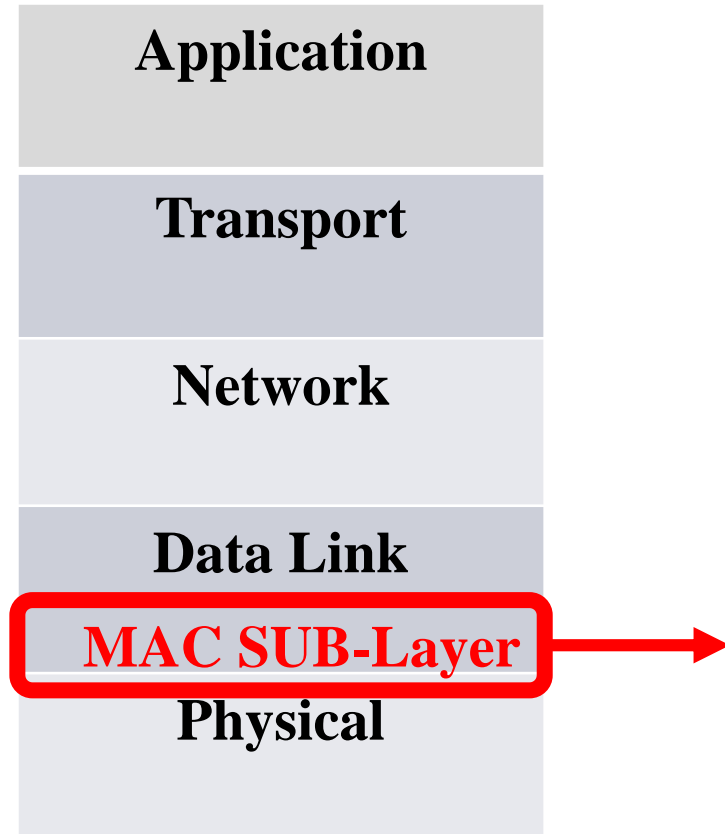
slot 5: 5, 7 (collision)

slot 6: 5 (success)

slot 7: 7 (success)



# Mac Sub-Layer

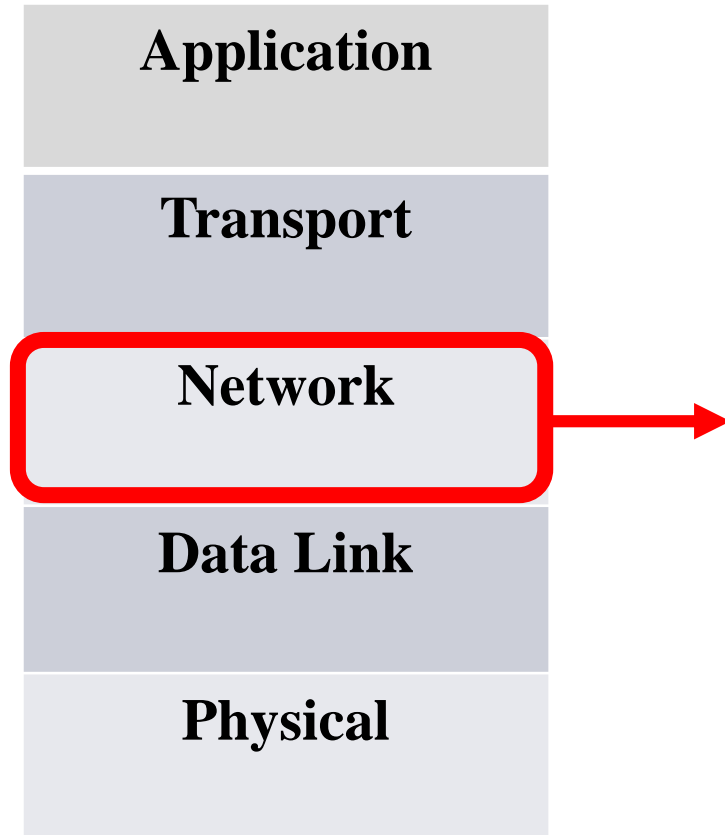


1. Contention
  - ALOHA
  - Carrier Sense Multiple Access (CSMA)
2. Collision Free
  - CSMA/CD – Binary Countdown
  - CSMA/CD - bit map
3. Limited Contention
  - CSMA/CD - Adaptive Tree Walk Protocol
4. MACA/MACAW (for Wireless LANs)



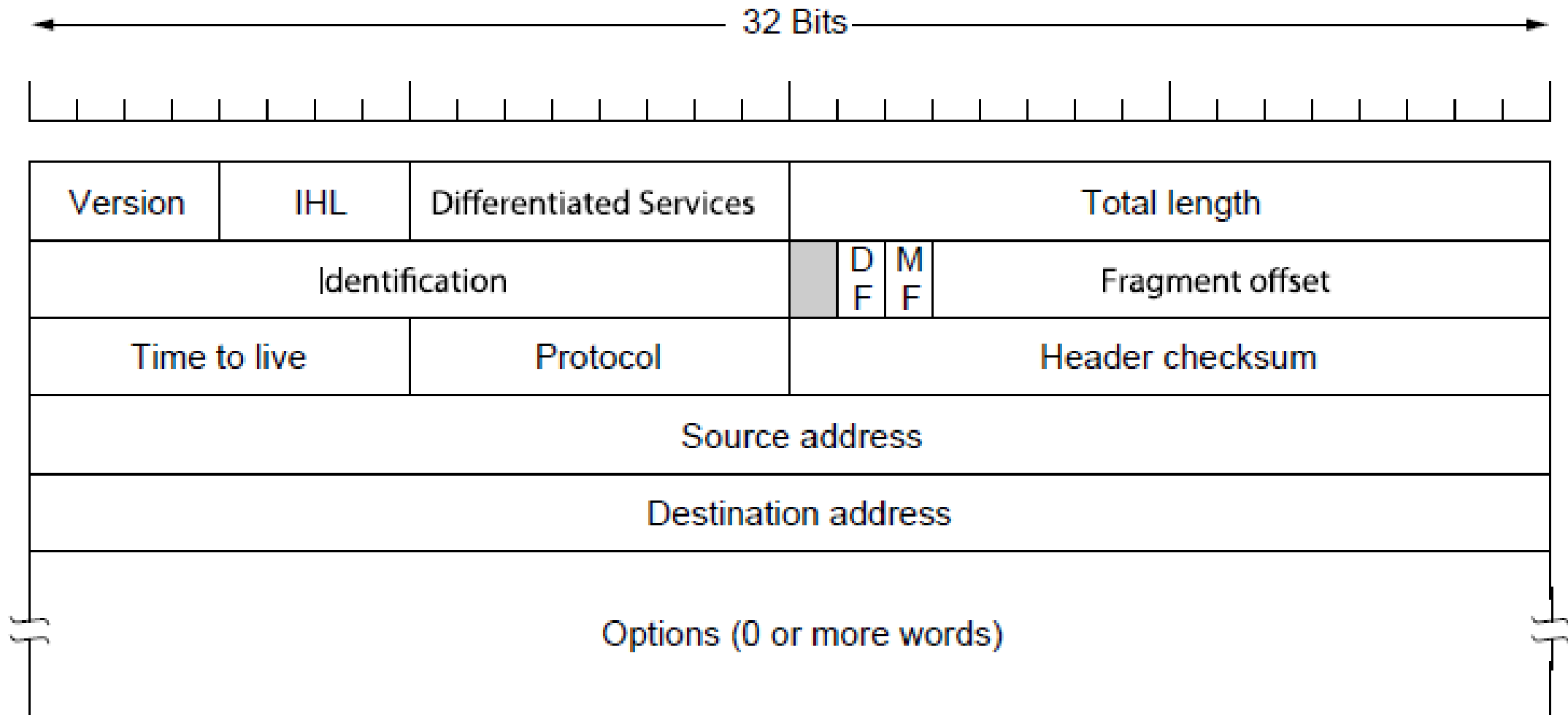
# Network Layer

# Network Layer



1. IPV4
2. IPV6
3. Routing Algorithms
  1. Non-adaptive
  2. Adaptive
  3. Hierarchical Routing
  4. Broadcasting routing
  5. Multicasting routing

# IP4 Datagram Structure





## Question5

Convert the IP address 11000001.01010010.11010010.00001111 to dotted decimal notation.

Ans. 193.82.210.15





## Question 6

Convert the IP address 240.68.10.10 to binary format

Ans. 1111 0000 . 0100 0100 . 0000 1010 . 0000 1010