

# 3 : Data Link Layer and MAC sublayer

IT4506 – Computer Networks

Level II - Semester 4

# Overview

- In this topic we discuss the design principles of datalink layer, error detections and error correction methods, protocols belong to datalink layer, and media access control sublayer.

# Intended Learning Outcomes

- At the end of this lesson, you will be able to;
  - Discuss the protocols in the datalink layer.
  - Describe the design decisions of the datalink layer protocols.
  - Explain the Error Detection and Error Correction methods.
  - Summarise the protocols used in the Medium Access Control Sublayer.

## List of sub topics

3.1 Design Issues

3.2 Error Detection and Error Correction

3.3 Elementary Data Link Protocols

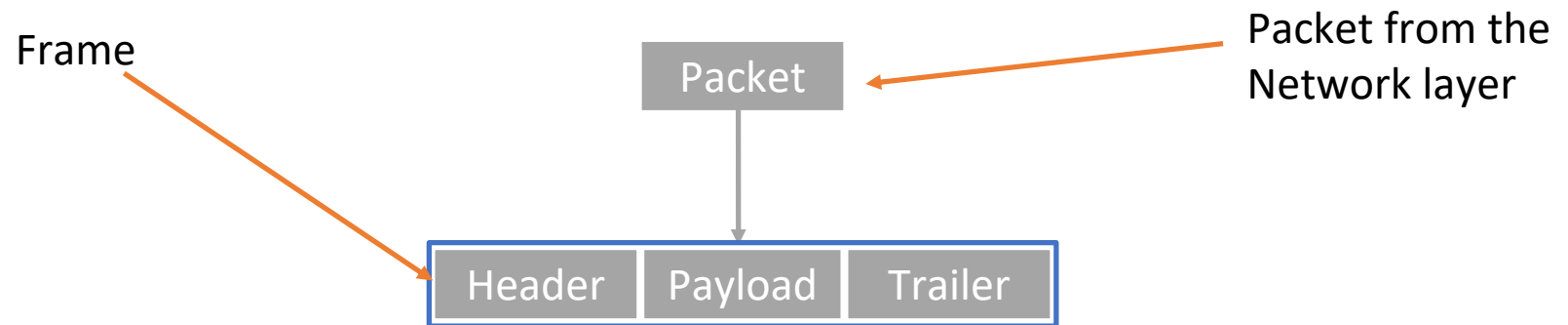
3.4 Medium Access Control Sublayer

# Datalink Layer - Design Issues

- The data link layer uses the services of the physical layer to send and receive bits over communication channels.
- Functions of Datalink layer
  - Providing a well-defined service interface to the network layer.
  - Dealing with transmission errors.
  - Regulating the flow of data so that slow receivers are not swamped by fast senders.

# Relationship with Packet and Frame

- The datalink layer obtain the packet from the Network layer and encapsulate into a "frame".
- The frame contains a frame header, payload which includes the packet obtained from the Network layer, and a trailer.



# Design Issues

- 1. Services Provided to the Network Layer
  - Transferring data from the network layer on the source machine to the network layer on the destination machine.
    - Unacknowledged connectionless service - source machine send independent frames to the destination machine without getting the acknowledgement from the destination machine. (Eg: Ethernet)
    - Acknowledged connectionless service - no logical connections used, but each frame sent is individually acknowledged. If the frame didn't received within the given period, source machine will retransmit. (Eg: WiFi)
    - Acknowledged connection-oriented service - source and destination machines establish a connection before any data are transferred. The data link layer guarantees that each frame sent is indeed received.

# Design Issues

- 2. Framing
  - The starting and the ending of the frame has to be defined properly so that the destination can recognise the frame properly.
  - Datalink layer will break up the bit stream into discrete frames and compute a short token called checksum for each frame.
  - This checksum is included in all the frames transmitted.
  - When the frame is received at the destination this checksum is recalculated.
  - If the recomputed checksum is different from the original checksum, the datalink layer knows that an error has occurred in the transmission.



# Design Issues

- 3. Error control
  - It has to be done to prevent duplication in frames.
  - If the sender receives a positive acknowledgement, the frame has been received safely.
  - If the sender receives negative acknowledgement or no acknowledgement within the time period, there is a problem with the frame. So the frame has to be retransmitted.

# Design Issues

- 4. Flow Control
  - Source machine should not transfer frames more than which can be received at the destination machine.
  - If it send more than the limit of the receiving buffer, frames will overflow.

# Error Detection and Error Correction

- Two basic strategies to deal with errors
  - Include enough redundant information to enable the receiver to deduce what the transmitted data must have been. (Error-correcting codes)
    - Hamming codes
  - Include only enough redundancy to allow the receiver to deduce that an error has occurred and have it request a retransmission. (Error-detecting codes)
    - Parity
    - Checksums

# Error Detection and Error Correction

- Parity
  - The parity bit is chosen so that the number of 1 bits in the codeword is even (or odd).
    - For example, when 1011010 is sent in even parity, a bit is added to the end to make it 1011010**0**. With odd parity 1011010 becomes 1011010**1**.
  - A code with a single parity bit has a distance of 2, since any single-bit error produces a codeword with the wrong parity.
  - This means that it can detect single-bit errors.

# Error Detection and Error Correction

- Parity
  - Consider 10110 as the data to be transmitted.
  - It has given 1 as the even parity bit and the resulting data will be 101101.
  - The receiver got an incorrect bit stream as 100101.
  - Receiving end calculate the parity.  $1 + 0 + 0 + 1 + 0 + 1 = 3$ , and  $3 \bmod 2 = 1$
  - It expect the answer to be 0. But instead it receives 1. Which means the transmitted data has an error.
  - Therefore the receiver ask the sender to resend the data.
  - The receive 101101 for the second time. It calculate the parity.  $1 + 0 + 1 + 1 + 0 + 1 = 4$  and  $4 \bmod 2 = 0$ , which means no errors in the transmission.

# Error Detection and Error Correction

- Checksum
  - Checksum for the data is calculated and send with the data frame.
  - At the receiving end the checksum is calculated and verifies that there is no errors in the data.

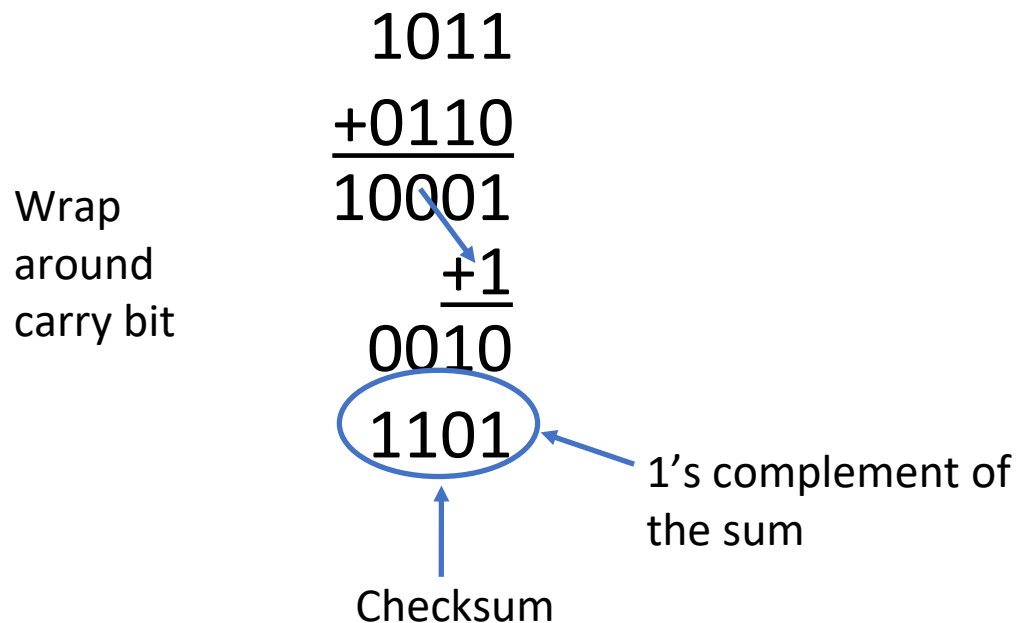
# Error Detection and Error Correction

- Checksum Example
  - Checksum (RFC1071)
    - Data is converted into series of 16 bit integers (it can be 8,16,32 and so on. For the example we consider 16 bit)
    - Calculate the sum of all 16 bit integers (Carry bit wrap around)
    - Take 1's complement of the final sum
  - Validate Checksum
    - Data is converted into series of 16 bit integers
    - Calculate the sum of all 16 bit integers (Carry bit wrap around)
    - Add the checksum to the final sum.
    - If the result is all 1's then the integrity of the data is verified.

<https://datatracker.ietf.org/doc/html/rfc1071>

# Error Detection and Error Correction

- Checksum 4 bit example
  - Lets use data 10110110
  - Brake it into two 4bit words 1011, 0110





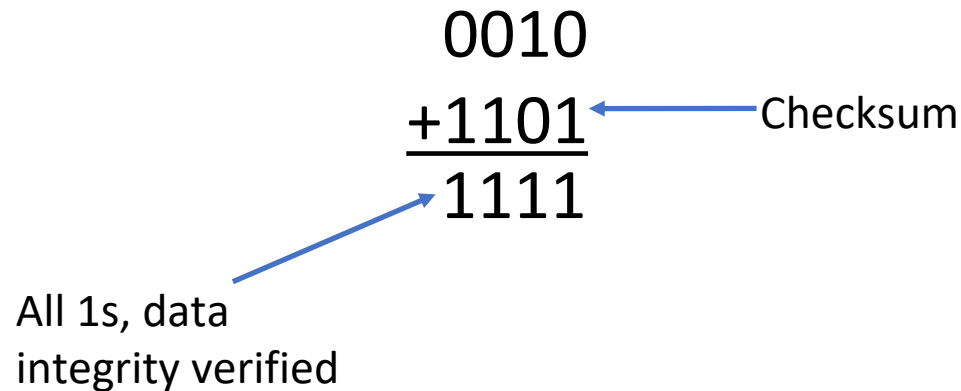
# Error Detection and Error Correction

- Checksum 4 bit example
  - Validate Checksum
    - Data sum - 0010

$$\begin{array}{r} 0010 \\ +1101 \\ \hline 1111 \end{array}$$

Checksum

All 1s, data integrity verified



# Error Correcting Codes

- The number of bit positions in which two codewords differ is called the **Hamming distance**.
- If two codewords are a hamming distance  **$d$**  apart, it will require  **$d$**  single-bit errors to convert one into the other.
- The error-detecting and error-correcting properties of a block code depend on its Hamming distance.
- To reliably detect  **$d$**  errors, you need a distance  **$d + 1$**  code because with such a code there is no way that  **$d$**  single-bit errors can change a valid codeword into another valid codeword.
- When the receiver sees an illegal codeword, it can tell that a transmission error has occurred.
- To correct  **$d$**  errors, you need a distance  **$2d + 1$**  code because that way the legal codewords are so far apart that even with  **$d$**  changes the original codeword is still closer than any other codeword.

# Error Correcting Codes

- Lets see what is an error is,
  - Codeword transmitted - 10001001
  - Codeword received - 10110001
- To determine how many bits differ need to take the XOR the two codewords and count the number of 1 bits in the result

$$\begin{array}{r} 10001001 \\ \text{XOR } \underline{10110001} \\ \hline 00111000 \end{array}$$

- In this example, 3 bits have changed in the transmission.

# Hamming Code

- Hamming codes use number of parity bits.
- The number of required parity bits can be calculated using the following equation (n is number of data bits, P is redundant bits)

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

- Locations of parity bits;
  - Parity bits are placed in positions numbered corresponding to the power of 2. (1, 2, 4, 8, 16....)
  - Parity bit at location 1 is responsible for all the locations has 1 in least significant position.
  - Parity bit at location 2 is responsible for all the locations has 1 in second least significant position and so on

# Hamming Code

- Example (Data - 11001 and even parity)
  - If  $n=5$

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

- Lets assume  $P=4$

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

- $P=4$  satisfies the equation.
  - Total number of bit;  $n + P = 5 + 4 = 9$

# Hamming Code

- Example
  - Locations of Data bits and Parity bits
  - Parity 1 should consider location 3, 5, 7, 9
  - Parity 2 should consider location 3, 6, 7
  - Parity 3 should consider location 5, 6, 7
  - Parity 4 should consider location 9

Bit location	9	8	7	6	5	4	3	2	1
Type	Data	Parity 4	Data	Data	Data	Parity 3	Data	Parity 2	Parity 1
Binary representation of location	1001	1000	0111	0110	0101	0100	0011	0010	0001
Data	1		1	0	0		1		
		1				1		0	1

- Data with parities - 111001101

# Elementary Data Link Protocols

- Utopian Simplex Protocol
  - Data are transmitted in one direction only
  - Both the transmitting and receiving network layers are always ready
  - Processing time can be ignored
  - Infinite buffer space is available
  - Communication channel between the data link layers never damages or loses frames
  - The sender is in an infinite while loop just pumping data out onto the line as fast as it can
  - The body of the loop consists of three actions
    - Go fetch a packet from the (always obliging) network layer
    - Construct an outbound frame
    - Send the frame
  - At the receiving end
    - Wait for the undamaged frame to receive
    - Removes the newly arrived frame from the hardware buffer
    - Then the data portion is passed on to the network layer
    - The data link layer settles back to wait for the next frame

# Elementary Data Link Protocols

- Simplex Stop-and-Wait Protocol for an Error-Free Channel
  - This is to preventing the sender from flooding the receiver with frames faster than is able to process them.
    - One solution is to build the receiver to be powerful enough to process a continuous stream of back-to-back frames.
    - A more general solution to this problem is to have the receiver provide feedback to the sender.
      - The receiver sends a little dummy frame (acknowledgement) back to the sender which, in effect, gives the sender permission to transmit the next frame.
  - Protocols in which the sender sends one frame and then waits for an acknowledgement before proceeding are called stop-and-wait



# Elementary Data Link Protocols

- Simplex Stop-and-Wait Protocol for a Noisy Channel
  - In the normal situation of a communication channel there can be errors. Frames may be either damaged or lost completely
  - Receiver hardware detects an error when calculating the checksum.
  - One solution is to use the acknowledgements. If the receiver does not send the acknowledgement within a time period, the sender can resend the data frame.
  - But what if the data frame received correctly, but the acknowledgement damaged or lost. The sender will retransmit the data frame and there will be duplicate data at the receiver's end.
  - As a solution the protocol can use sequence numbers.

# Overview of Sliding Window Protocols

- In the previous three protocols, data transmitted in one direction. But in most practical situations there is a need to transmit data in both directions.
- One way of achieving full-duplex data transmission is to run two instances of one of the previous protocols, each using a separate link for simplex data traffic.
- Each link is then comprised of a **forward** channel (for data) and a **reverse** channel (for acknowledgements). In both cases the capacity of the reverse channel is almost entirely wasted.
- A better idea is to use the same link for data in both directions.
  - Simplex Stop-and-Wait Protocol for an Error-Free Channel and Simplex Stop-and-Wait Protocol for a Noisy Channel was already being used to transmit frames both ways.
  - In this model the data frames from A to B are intermixed with the acknowledgement frames from A to B.
  - By looking at the kind field in the header of an incoming frame, the receiver can tell whether the frame is data or an acknowledgement.

# Overview of Sliding Window Protocols

- When a data frame arrives, instead of immediately sending a separate control frame, the receiver restrains itself and waits until the network layer passes it the next packet.
- The acknowledgement is attached to the outgoing data frame using the ack field in the frame header.
- The acknowledgement gets a free ride on the next outgoing data frame.
- The technique of temporarily delaying outgoing acknowledgements so that they can be hooked onto the next outgoing data frame is known as **piggybacking**.
  - How long should the data link layer wait for a packet onto which to piggyback the acknowledgement?
    - It wait for a fixed number of milliseconds. If a new packet arrives quickly, the acknowledgement is piggybacked onto it. Otherwise, if no new packet has arrived by the end of this time period, the data link layer just sends a separate acknowledgement frame.

# Overview of Sliding Window Protocols

- In sliding window protocols, each outbound frame contains a sequence number, ranging from 0 up to some maximum.
- The maximum is usually  $2^n - 1$  so the sequence number fits exactly in an n-bit field.
- The essence of all sliding window protocols is that at any instant of time, the sender maintains a set of sequence numbers corresponding to frames it is permitted to send. These frames are said to fall within the **sending window**.
  - The sequence numbers within the sender's window represent frames that have been sent or can be sent but are as yet not acknowledged.
- The receiver also maintains a **receiving window** corresponding to the set of frames it is permitted to accept.
  - Any frame falling within the window is put in the receiver's buffer. When a frame whose sequence number is equal to the lower edge of the window is received, it is passed to the network layer and the window is rotated by one.

# Medium Access Control Sublayer

- Overview of channel allocation problem
  - How to allocate a single broadcast channel among competing users?
    - Static Channel Allocation
      - Frequency Division Multiplexing (FDM) - If there are  $N$  users, the bandwidth is divided into  $N$  equal-sized portions, with each user being assigned one portion.
      - Time Division Multiplexing (TDM) - Allocate each user every  $N$ th time slot, if a user does not use the allocated slot, it would just lie fallow.
      - A static allocation is a poor fit to most computer systems, in which data traffic is extremely bursty, often with peak traffic to mean traffic ratios of 1000:1. Consequently, most of the channels will be idle most of the time.
    - Dynamic Channel Allocation

# Medium Access Control Sublayer

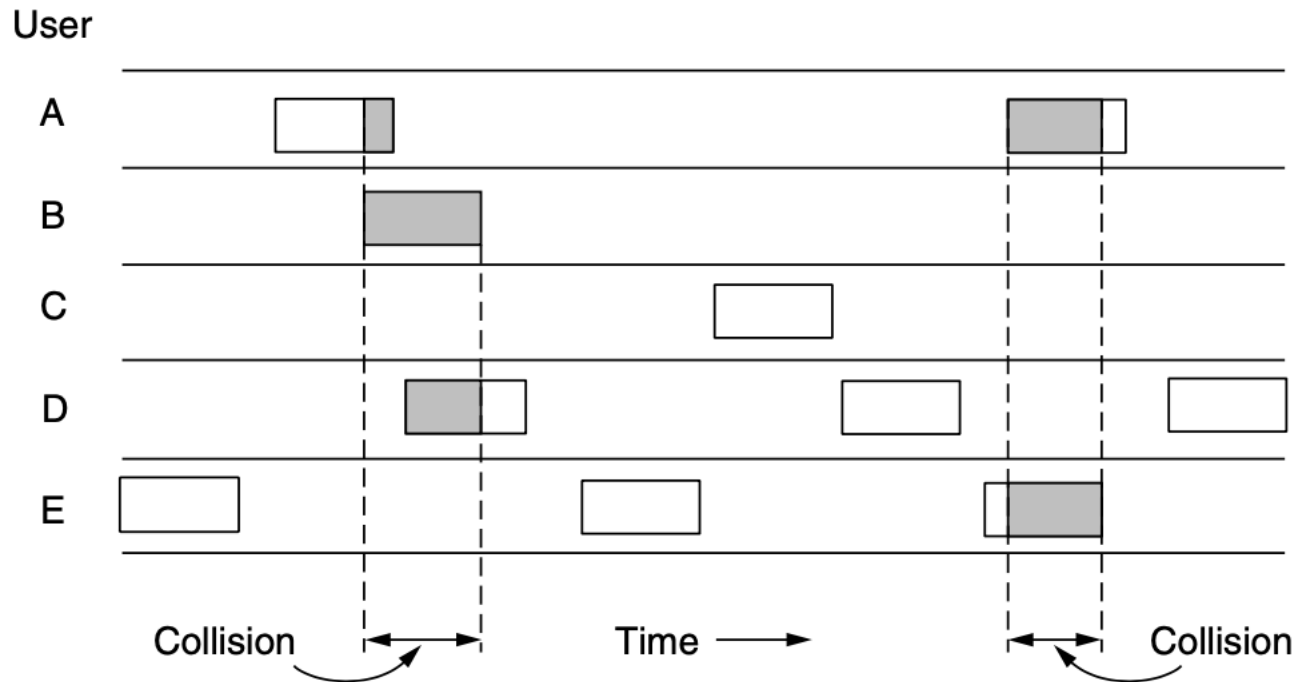
- Overview of channel allocation problem
  - How to allocate a single broadcast channel among competing users?
    - Dynamic Channel Allocation - Assumptions
      - Independent Traffic - The model consists of  $N$  independent stations, each with a program or user that generates frames for transmission.
      - Single Channel - A single channel is available for all communication.
      - Observable Collisions - If two frames are transmitted simultaneously, they overlap in time and the resulting signal is garbled.
      - Continuous or Slotted Time - Time may be assumed continuous, in which case frame transmission can begin at any instant or time may be slotted or divided into discrete intervals.
      - Carrier Sense or No Carrier Sense - stations can tell if the channel is in use before trying to use.

# Multiple Access Protocols

- Pure ALOHA
  - Let users transmit whenever they have data to be sent.
  - There will be collisions, of course, and the colliding frames will be damaged.
  - Senders need some way to find out whether there is an error occurred.
    - In the ALOHA system, after each station has sent its frame to the central computer, this computer rebroadcasts the frame to all of the stations.
    - A sending station can thus listen for the broadcast from the hub to see if its frame has gotten through.
    - If the frame was destroyed, the sender just waits a random amount of time and sends it again.

# Multiple Access Protocols

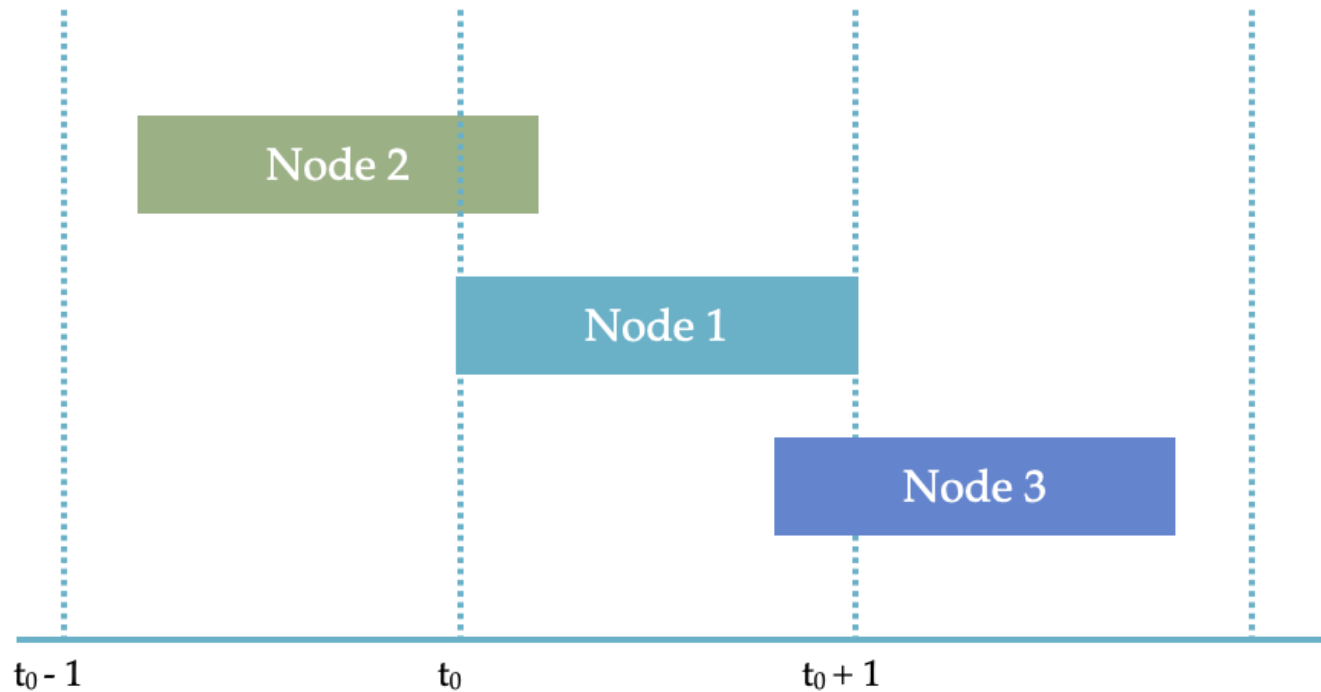
- Pure ALOHA
  - When five nodes are using the same medium.





# Multiple Access Protocols

- Efficiency of Pure ALOHA



# Multiple Access Protocols

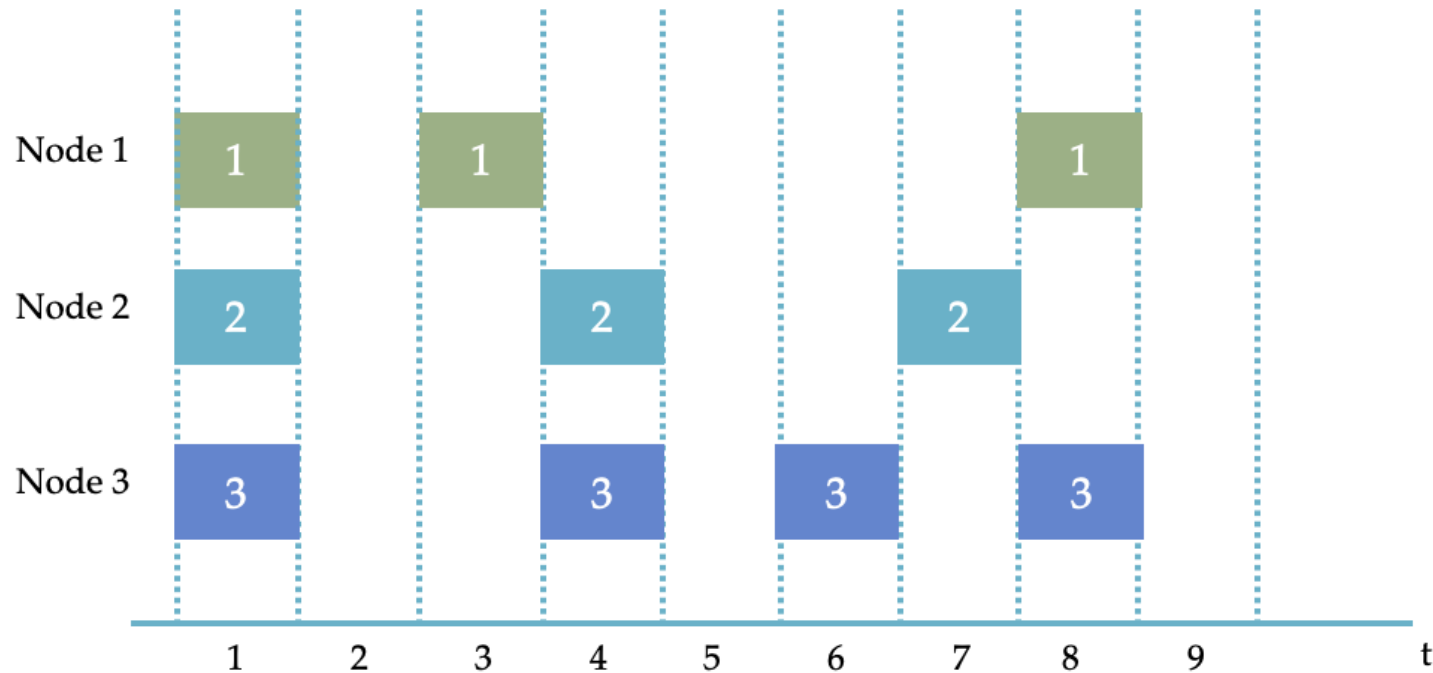
- Efficiency of Pure ALOHA
  - New frame generation is modelled by poisson distribution with mean of  $N$
  - Old and new frames modelled by poisson distribution with mean of  $G$   
(Mean number of frames generated in frame time)
  - Throughput:  $S = GP_0$
  - Probability of  $k$  frames generated during a frame time is  $Pr[k] = \frac{G^k e^{-G}}{k!}$
  - Probability of zero frames  $e^{-G}$
  - Mean number of frames generated in two frame times is  $2G$
  - Probability of no frames generated in vulnerable period:  $P_0 = e^{-2G}$
  - Therefore throughput  $S = Ge^{-2G}$
  - Maximum in  $G = 0.5$ , which is  $S = \frac{1}{2e}$

# Multiple Access Protocols

- Slotted ALOHA
  - Unlike pure ALOHA , in slotted ALOHA a station is not permitted to send when-ever the user have data to be sent. Instead, it is required to wait for the beginning of the next slot.
  - The continuous time ALOHA is turned into a discrete time one.
  - This halves the vulnerable period.

# Multiple Access Protocols

- Efficiency of Slotted ALOHA

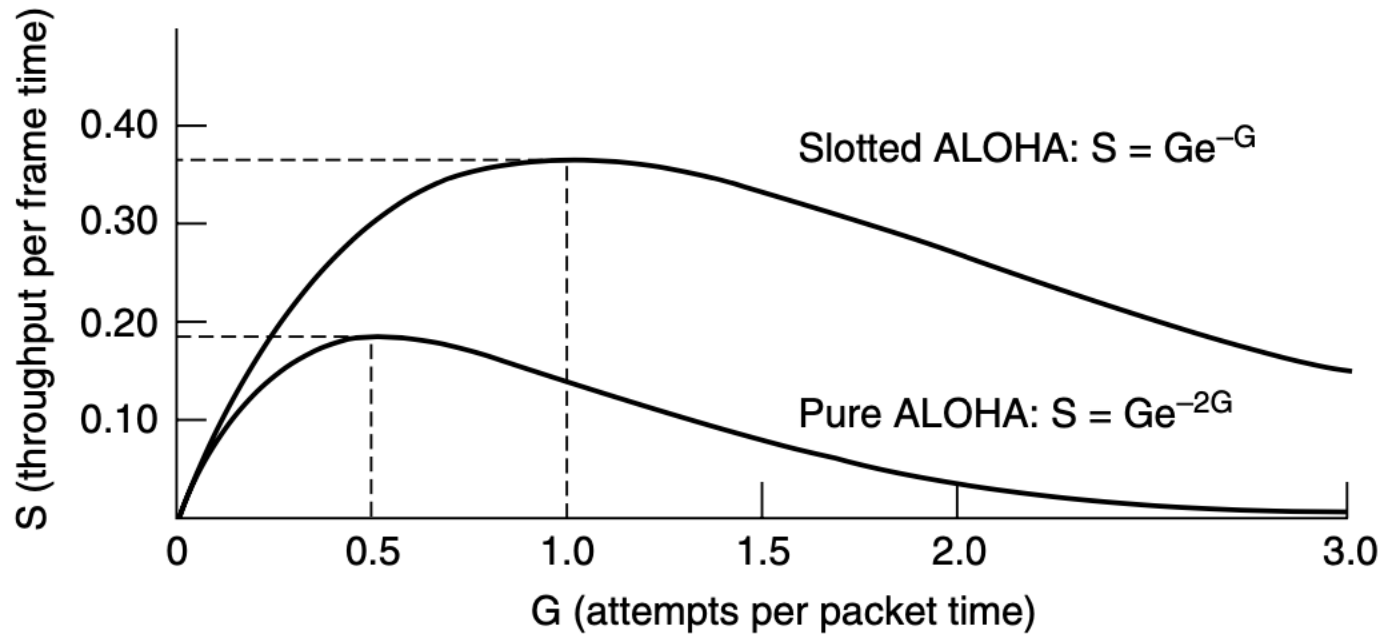


# Multiple Access Protocols

- Efficiency of Slotted ALOHA
  - New frame generation is modelled by poisson distribution with mean of  $N$
  - Old and new frames modelled by poisson distribution with mean of  $G$   
(Mean number of frames generated in frame time)
  - Throughput  $S = GP_0$
  - Probability of  $k$  frames generated during a frame time is  $Pr[k] = \frac{G^k e^{-G}}{k!}$
  - Probability of zero frames  $e^{-G}$
  - Mean number of frames generated in one frame times is  $G$
  - Probability of no frames generated in vulnerable period  $P_0 = e^{-G}$
  - Therefore throughput  $S = Ge^{-G}$
  - Maximum in  $G = 1$ , which is  $S = \frac{1}{e}$

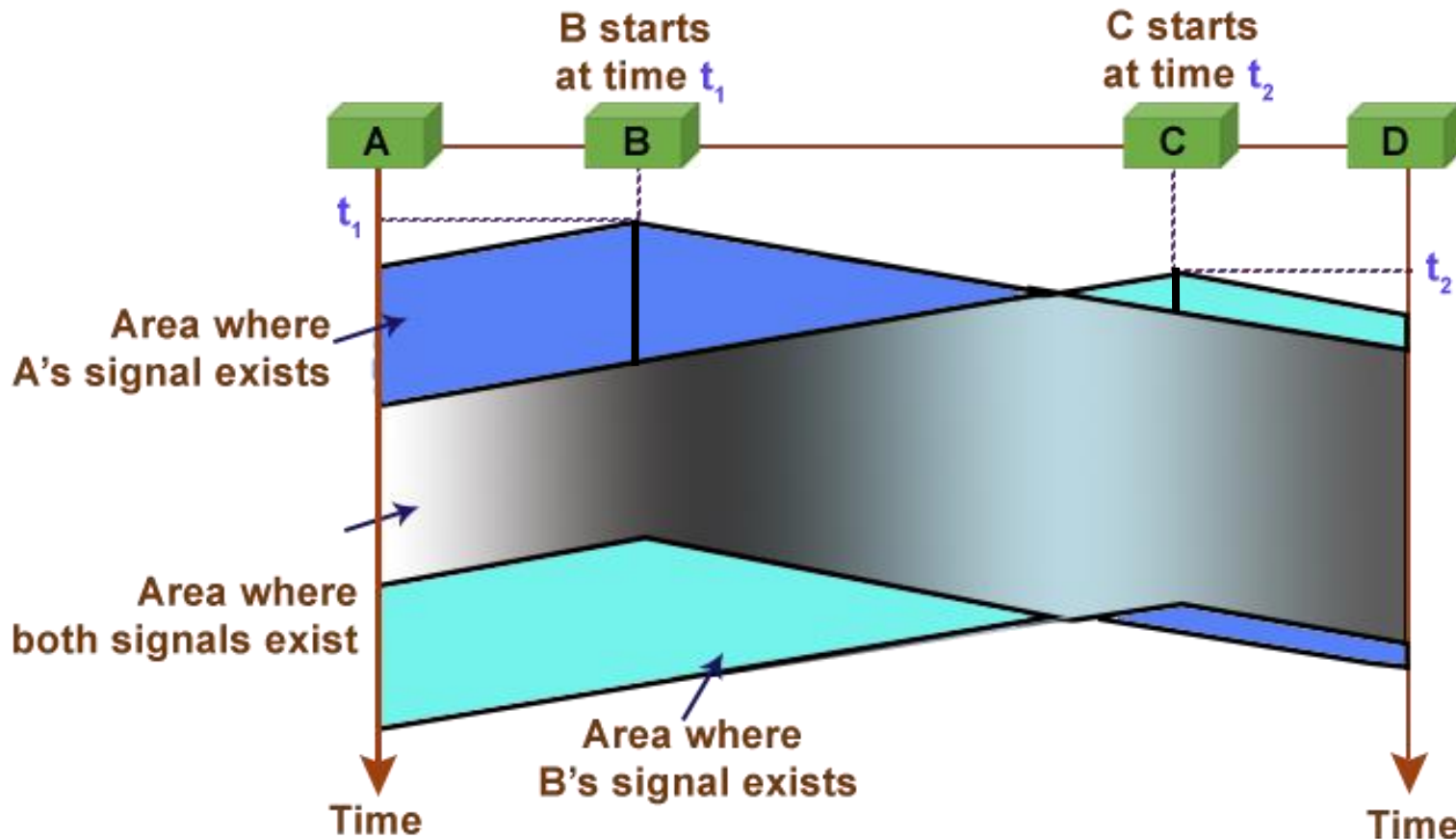
# Multiple Access Protocols

- Pure ALOHA vs Slotted ALOHA



# Carrier Sense Multiple Access Protocols

- Protocols in which stations listen for a carrier and act accordingly are called carrier sense protocols.



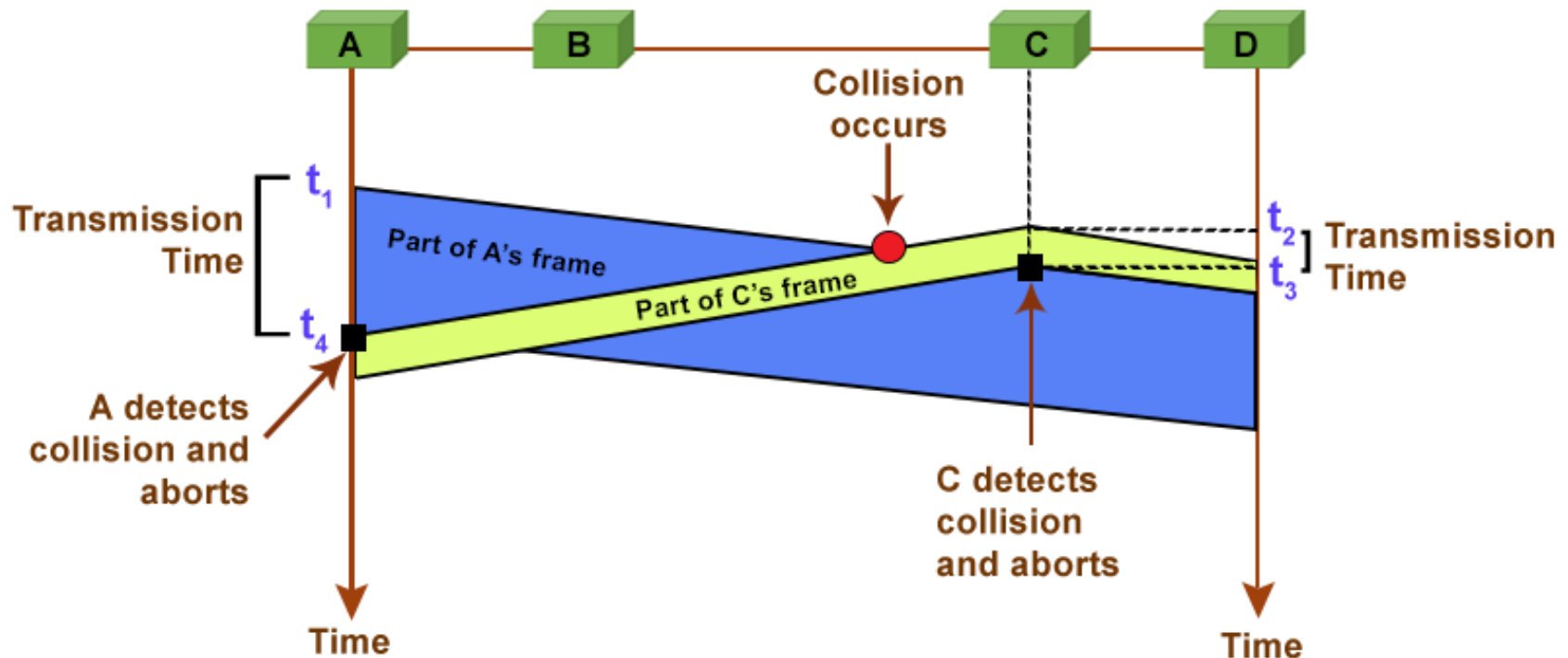
# Carrier Sense Multiple Access Protocols

- 1 persistent
  - When a station has data to send, it first listens to the channel to see if anyone else is transmitting at that moment.
  - If the channel is idle, the stations sends its data.
  - If the channel is busy, the station just waits until it becomes idle.
  - The protocol is called 1-persistent because the station transmits with a probability of 1 when it finds the channel idle.
- Non persistent
  - A station senses the channel when it wants to send a frame, and if no one else is sending, the station begins to transmit.
  - If the channel is already in use, the station does not continually sense, it waits a random period of time and then repeats the algorithm.
- P persistent
  - It applies to slotted channels
  - When a station becomes ready to send, it senses the channel. If it is idle, it transmits with a probability  $p$ .
  - With a probability  $q = 1 - p$ , it defers until the next slot.



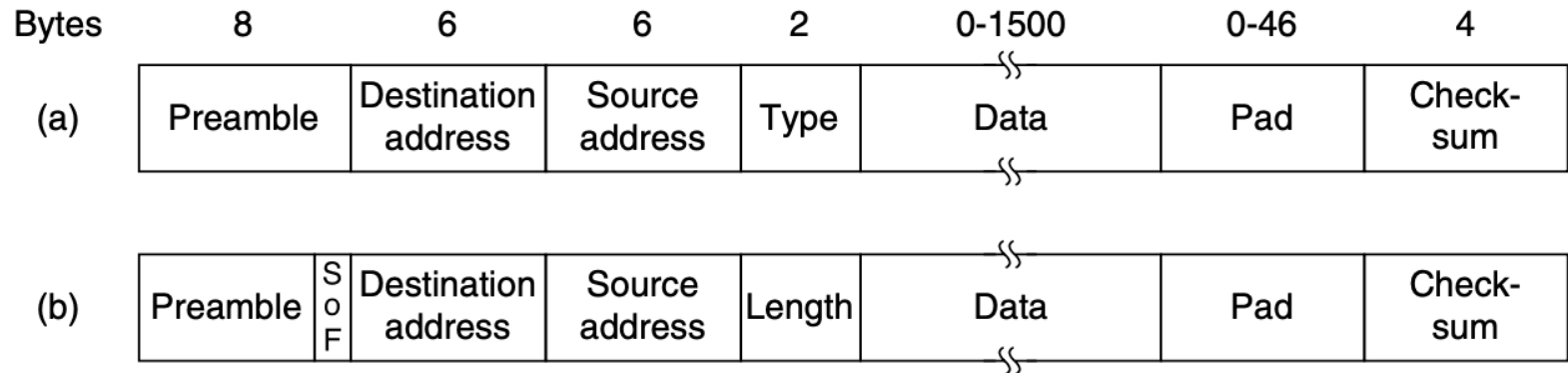
# Carrier Sense Multiple Access Protocols

- Carrier Sensed Multiple Access/ Collision Detection (CSMA/CD)
  - Stations will detect the collision and stop transmitting.
  - CSMA/CD, is the basis of the classic Ethernet LAN.



# Ethernet

- Frame format



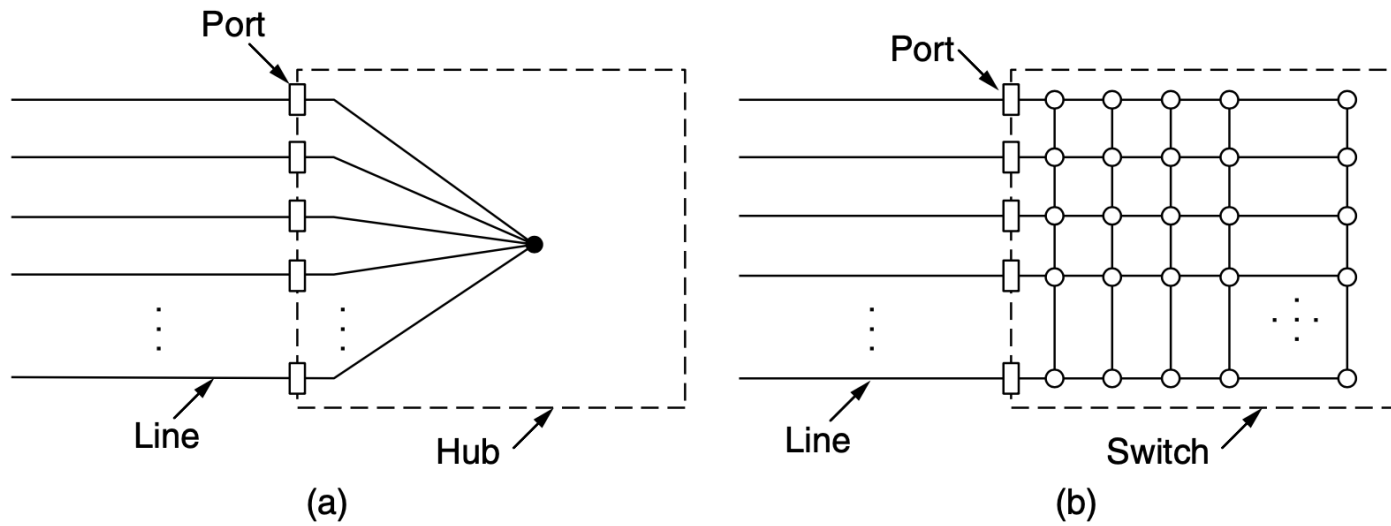
- (a) Ethernet (DIX)
- (b) IEEE 802.3

# Ethernet

- Classic Ethernet MAC Sublayer Protocol
  - **Preamble** of 8 bytes, each containing the bit pattern 10101010 (with the exception of the last byte, in which the last 2 bits are set to 11). This last byte is called the Start of Frame delimiter for 802.3.
  - **Destination and Source Address** : each 6 bytes long. These addresses are globally unique, assigned centrally by IEEE to ensure that no two stations anywhere in the world have the same address. To do this, the first 3 bytes of the address field are used for an OUI (Organizationally Unique Identifier). The manufacturer assigns the last 3 bytes of the address
  - **Type or Length**: In Ethernet, **Type** tell the receiver what to do with the frame. In IEEE 802.3, **Length** is the length of the frame.
  - **Pad**: field is used to fill out the frame to the minimum size
  - **Checksum**: CRC is an error-detecting code

# Ethernet

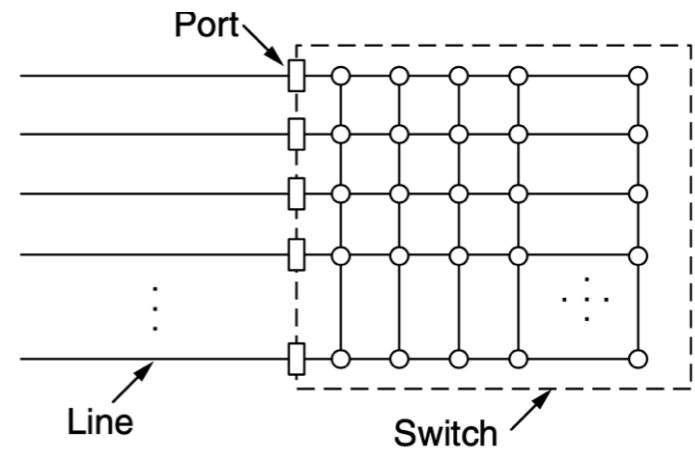
- Switched Ethernet



- (a) Hub
- (b) Switch

# Ethernet

- Switched Ethernet
  - Architecture of Hub is logically equivalent to the single long cable of classic Ethernet. When number of nodes increases each node gets a decreasing share of the fixed capacity.
  - Solution for this is the **Switched Ethernet**.
  - The Switch contains a high-speed backplane that connects all of the ports.



# Ethernet

- Switched Ethernet
  - Switches only output frames to the ports for which those frames are destined.
  - When the switch received an Ethernet frame, it checks the Ethernet address and send the frame to the designated port.
  - In the switch,
    - There is no collisions
    - Switch sends multiple frames simultaneously.

# Ethernet

- Fast Ethernet
  - Ethernet standard was IEEE 802.3 and the average speed was around 10 Mbps. But with Fast Ethernet (IEEE 802.3u) speed increased over 100Mbps.
  - Twisted Pair (100Base-T4, 100Base-TX) and Fiber optics (100Base-FX) cables are used.
- Gigabit Ethernet
  - Standard is IEEE 802.3ab
  - Transfer speed up to 1000Mbps (1Gbps)
  - Twisted Pair (1000Base-CX two pairs of STP, 1000Base-T four pairs of UTP) and Fiber optics (1000Base-SX, 1000Base-LX) cables are used.
- 10-Gigabit Ethernet
  - Supports 10Gbps speeds.
  - Twisted Pair (10GBase-CX4 four pairs of twinax, 10GBase-T four pairs of UTP) and Fiber optics (10GBase-SR, 10GBase-LR, 10GBase-ER) cables are used.

# Ethernet

- Activity - Retrospective on Ethernet
  - Discuss the expansion of Ethernet technologies from the begin.

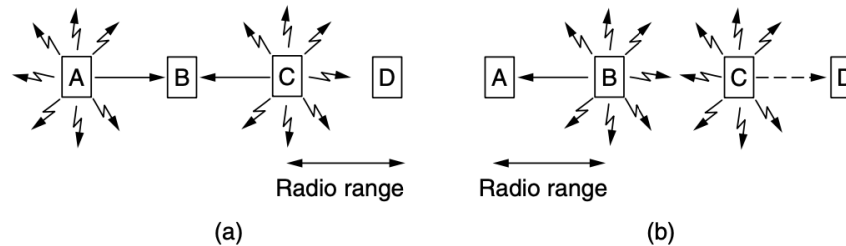


# Wireless LAN

- Wireless LAN Protocols
  - Commonly available on buildings with Wireless Access Points. These access points are connected with copper wires.
  - When comparing wired LANs, wireless LANs cannot identify collisions.
  - When Wireless LANs transmit frames, it received by all the stations.
  - Wireless LANs use CSMA, to listen the medium and send frames if the medium is not used by other stations.
  - There are some issues with the Wireless LANs
    - Hidden terminal problem
    - Exposed terminal problem

# Wireless LAN

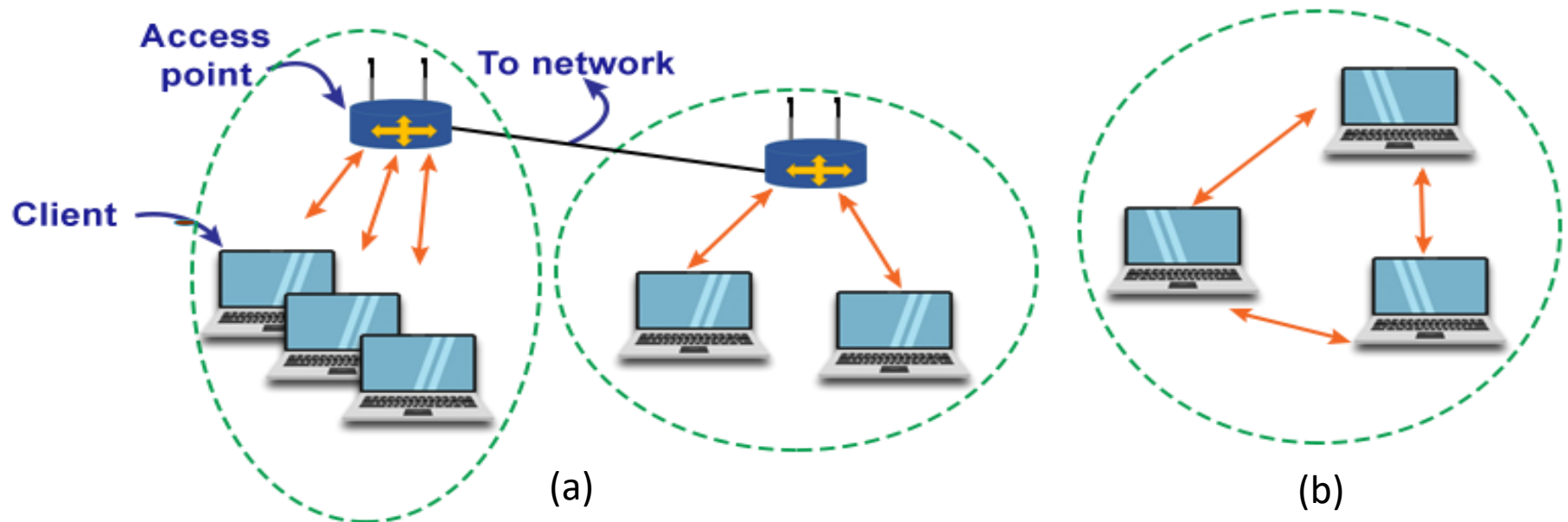
- Wireless LAN Protocols



- Hidden terminal problem
  - Consider figure (a). A and C want to transmit to B. But C is not within the range of A. Therefore, if A is transmitting, C cannot sense it though it sense the medium. If C start transmission, collisions may occur at B. This is hidden terminal problem.
- Exposed terminal problem
  - Consider figure (b). B need to transmit to A and C need to transmit to D. If C sense the medium while B is communicating with A, C sense the medium is already in use. This is exposed terminal problem. Though C and D can do the communication, it is prohibited because of the exposed terminal problem.

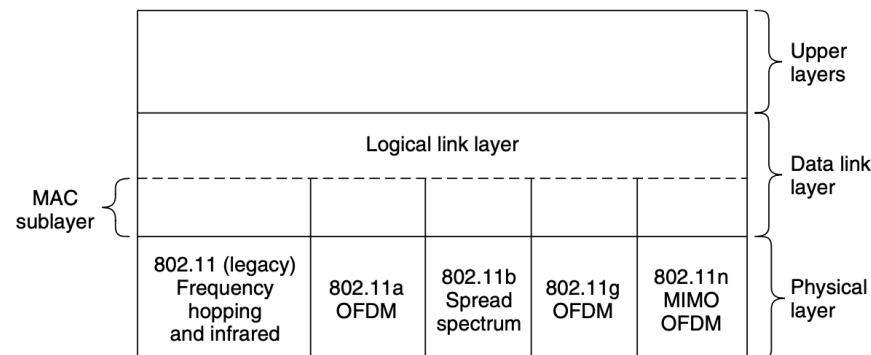
# Wireless LAN

- The 802.11 Architecture and Protocol Stack
  - 802.11 Architecture has two modes
    - Infrastructure mode (a)
    - Ad-hoc mode (b)



# Wireless LAN

- The 802.11 Architecture and Protocol Stack
  - In 802.11 Architecture data link layer split into two or more sublayers
    - MAC (Medium Access Control) sublayer - determine how the channel is allocated and which station gets the chance to transmit next.
    - LLC (Logical Link Control) sublayer - works in between the Network layer and hide the differences of 802 variates to the upper layers.



# Wireless LAN

- The 802.11 Physical Layer
  - All the 802.11 variants uses short-range radios to transmit signals in either the 2.4-GHz or the 5-GHz ISM frequency bands
  - 802.11b supports rates of 1, 2, 5.5, and 11 Mbps.
  - 802.11a, supports rates up to 54 Mbps in the 5-GHz ISM band.
  - 802.11n, supports rates up to 600 Mbps. It use MIMO (Multiple Input Multiple Output) which allows to use multiple antennas.
  - There are some other standards such as 802.11ac, 802.11ad, 802.11af, 802.11ah and 802.11ax.

# Wireless LAN

- The 802.11 MAC Sublayer Protocol
  - Different from Ethernet due to two factors
    - Communication is nearly half duplex, which means it cannot transmit and listen for noise bursts at the same time on a single frequency.
      - Therefore to avoid collisions 802.11 use CSMA/CA (CSMA with Collision Avoidance)
  - Transmission ranges of different stations may be different.
    - Due to this hidden terminal problem and exposed terminal problem occurs.
      - Protocol use Request to send (RTS) and Clear to send (CTS) frames as a solution for the problems.

# Data Link Layer Switching

- Many organisations have multiple LANs
- It is possible to connect these LANs together by using a device called **Bridge** (Switch).
- Bridge is operating in the Datalink layer.
  - Therefore, it examine the datalink address and forward the frames to the relevant destination.
- Uses of Bridges;
  - It can use to connect two different LANs
  - Or it can used to split single LAN in to separate LANs
- There is a table in the bridge. This table contain the details of the destination and the port attached to it. When the bridge is first plugged in, this table is empty and it will gradually fill the table by using the flooding algorithms.

# Data Link Layer Switching

- To increase the reliability, redundant links are setup between bridges.
  - This can also introduce some problems, such as loops.
  - To reduce this issue **Spanning Tree Protocol (STP)** is used.
  - STP ignore some potential connections to construct the loop free topology.



# Data Link Layer Switching

- Which device is in which layer
  - Application Layer - Application Gateway
  - Transport Layer - Transport Gateway
  - Network Layer - Router
  - Datalink Layer - Bridge/ Switch
  - Physical Layer - Hub/ Repeater

# Data Link Layer Switching

- Virtual LANs
  - There can be some situations where the network of the organisation has to be changed.
    - The network devices and wiring need to be changed and rewired. Which is difficult and not practical.
  - The solution is Virtual LAN.
    - There is a configuration table in the bridge to setup the VLAN.
    - It defines which port belongs to which VLAN.