CSMA = Carrier Sensing + Multiple Access

Carrier Sense - Devices "listen" to the network (carrier sensing) before transmitting.
Multiple Access – Broadcast link used in a network shared by multiple nodes that can both send and receive data.

The CSMA method was developed to minimize the chance of collision and, therefore, increase the performance.
The chance of collision can be reduced if a station senses the medium before trying to use it.

Carrier sense multiple access (CSMA) requires that each station first listen to the medium before sending. In other words, CSMA is based on the principle "sense before transmit" or "listen before talk." CSMA can reduce the possibility of collision, but it cannot eliminate it.

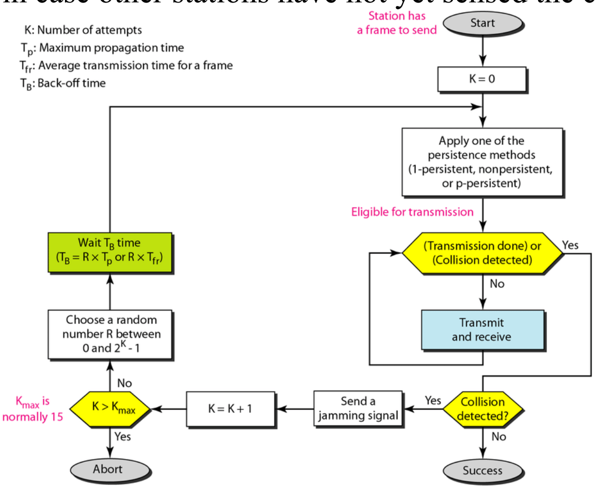
The vulnerable time for CSMA is the propagation time Tp:
Propagation time - **Propagation time** (or **propagation delay**) is the amount of time it takes for a signal (data) to **travel from the sender to the receiver** through a transmission medium.

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

Commonly used in Ethernet networks with hubs. It is difficult to detect collisions in wireless networks such as radio networks hence Collision detection is usually applied in wired networks.

How it works:
Devices "listen" to the network (carrier sensing) before transmitting.
If the channel is clear, they transmit.
However, if two devices transmit at the same time, a collision occurs.
The collision is detected, and both devices wait for a random backoff time before trying again.

Another important aspect is the sending of a short jamming signal that enforces the collision in case other stations have not yet sensed the collision.



Energy level of a channel is used to monitor the current status of a channel. The level of energy in a channel can have three values: zero, normal, and abnormal.

- At the zero level, the channel is idle.
- At the normal level, a station has successfully captured the channel and is sending its frame.
- At the abnormal level, there is a collision and the level of the energy is twice the normal level.

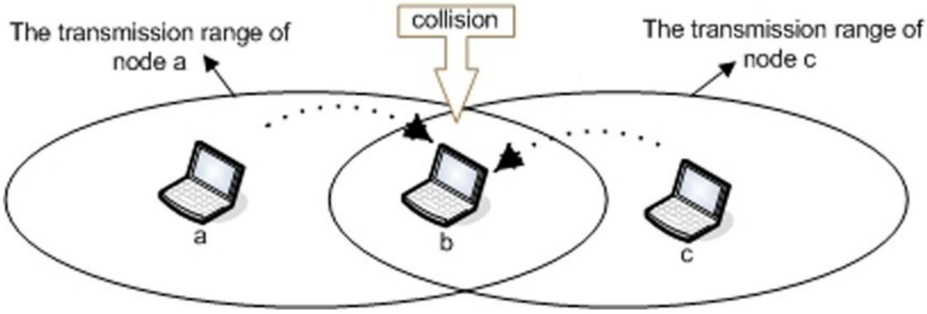
CSMA/CA

Commonly used in wireless networks. We can try to avoid collisions in wireless networks but detecting collisions is not easy.

Unlike CSMA/CD (used in wired Ethernet), which detects collisions, **CSMA/CA tries to avoid them before they happen**.

In wireless networks (like Wi-Fi), devices cannot easily detect collisions due to:

- Signal fading
- Hidden station problems



How CSMA/CA Works (Steps):

1. Carrier Sensing:
   - The device listens to the channel.
   - If it's idle, it proceeds. If it's busy, it waits.

2. Random Backoff:
   - Even if the channel is idle, the device waits for a random backoff time to avoid simultaneous transmission by multiple devices.

3. Optional RTS/CTS Handshake (Request to Send / Clear to Send):
   - Device sends RTS to the receiver.
   - Receiver replies with CTS if it's ready.
   - This informs other devices to wait, reducing hidden node problems.

4. Data Transmission:
   - Once CTS is received, the device sends the actual data.

5. Acknowledgment (ACK):
   - Receiver sends an ACK to confirm successful receipt.
   - If no ACK is received, the sender retries.

Exercises

1. A library with 15 computers connected via a hub-based wired network experiences slow data transfers and errors during peak hours when students access online databases. The network uses a protocol that checks if the channel is clear before transmitting, but simultaneous transmissions cause collisions, requiring retransmissions.

Identify the protocol causing the issue and propose a better protocol to prevent collisions and improve performance.

Solution:

Protocol described: **CSMA/CD** (Carrier Sense Multiple Access with Collision Detection), commonly used in Ethernet networks (wired networks) with hubs.

How a hub works in a hub based network: When a computer sends data to the hub, the hub **does not know the destination**. The hub **forwards (broadcasts)** the data to **all ports**, except the one it received the data from. All connected devices receive the signal, but **only the intended recipient** accepts and processes the data; others ignore it.

Upgrade from a hub to a switch and continue using Ethernet in full-duplex mode (which does not require CSMA/CD anymore).

Benefits: Eliminates collisions, Improves data transfer speeds, Enhances reliability and scalability, Maintains compatibility with existing Ethernet protocols.

2. "Checksum" error detection code checks sum of all letters.

   Let

   A ==> 1
   B ==> 2
   C ==> 3

what is "Checksum" code for "HELLO"?

Solution: 8+5+12+12+15=52

3. A system uses even parity. Tick (v) to show whether the following three bytes have been transmitted correct incorrectly.

Received Byte	Byte transmitted correctly	Byte transmitted incorrectly
11001000		x
01111100		x
01101001	x	

Solution:

Parity check - 1-bit check value - Based on the number of 1's in the message

- Even parity: number of 1's remains even
- Odd parity: number of 1's remains odd

4.

(b) A parity byte is used to identify which bit has been transmitted incorrectly in a block of data.

The word "F L O W C H A R T" was transmitted using nine bytes of data (one byte per character). A tenth byte, the parity byte, was also transmitted.

The following block of data shows all ten bytes received after transmission. The system uses **even parity** and column 1 is the parity bit.

	letter	column 1	column 2	column 3	column 4	column 5	column 6	column 7	column 8
byte 1	F	1	0	1	0	0	1	1	0
byte 2	L	1	0	1	0	1	1	0	0
byte 3	O	1	0	1	0	1	1	1	1
byte 4	W	1	0	1	1	0	1	1	1
byte 5	C	1	0	1	0	0	0	1	1
byte 6	H	0	0	1	0	1	0	0	0
byte 7	A	0	0	1	0	0	1	0	1
byte 8	R	1	0	1	1	0	0	1	0
byte 9	T	1	0	1	1	0	1	0	0
parity byte		1	0	1	1	1	1	1	0

(i) **One** of the bits has been transmitted incorrectly.

Write the byte number and column number of this bit:

Byte numbe. **Byte 7 (A)** is the only row where the total number of 1s is **odd** → this row has the incorrect bit.

Column number Columns **4** and **6** have **odd parity** → one of them is the column with the error.

[2]

Step 1: Check Each Row for Even Parity
Step 2: Check Each Column (2–8) for Even Parity
Step 3: Find the Intersecting Byte and Column

- The row with an error: Byte 7
- The columns with errors: Column 4 and Column 6

We now look at Byte 7 and check values at columns 4 and 6:

- Column 4: 0
- Column 6: 1

We flip each bit to see which one fixes both the row and column parity:

- Byte 7 total 1s: was 3 → now 4 (even)
- Column 4 total 1s: was 3 → now 4 (even)
- Column 6 total 1s: was 6 → now 5 (odd)

Final answer is Byte 7 and Column 4