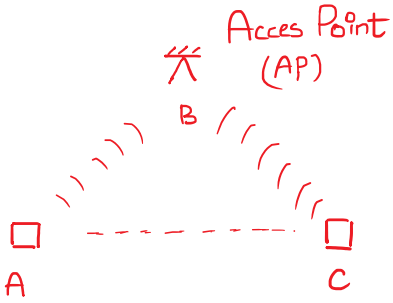
Wireless LANS – DLL (Wifi)

The protocols developed earlier of Ethernet such as CSMA-CD can't be directly applied to the case of Wifi.

LAN – 10s of devices within 100m → Wifi

WAN – Large gap between devices of order 1km.

CSMA-CD is a wired LAN protocol.



In this case, we have already discussed that the energy of the wave falls off as $1/d^{\alpha}$ where $(2 < \alpha < 5)$. Even if the SNR is reasonable at B, it might be less than 1 at C.

– This causes carrier sense to fail as the attenuation is too large

Also, collision detection fails because the change in energy is very faint.

- It is possible for carrier sense to work if A, C are close by but collision detection is a lost cause due to the large attenuation

We thus still use carrier sense but collision detection needs to be remodelled to better suit these limitations.

- Wifi Collision Detection — Ack frames

A first does carrier sensing for a while and it sends the frame after a set time. It waits for an Ack frame for a while, and it assumes collision if time-out occurs. It retransmits the data later following exponential backoff.

This method has other problems —

1) Hidden Terminal Problem — Virtual Carrier Sensing

- In above case, if B can hear A, C but A, C can't hear each other. This is the problem already discussed in carrier sensing.
- This would be extremely detrimental if the frame was large in size. The solution used by Wifi is called Virtual Carrier Sensing (VCS). We use the AP (B) as an intermediary.

- Suppose that in the same example, A wishes to send a frame to B. It first sends a Request to Send (RTS) to B and if B is free, it sends out a Clear to Send (CTS). Once received, A starts to transmit the data.

Notice that C does not receive the RTS, but it can hear the CTS. After sensing the CTS, C will stay silent even if it has data that needs to be transferred.

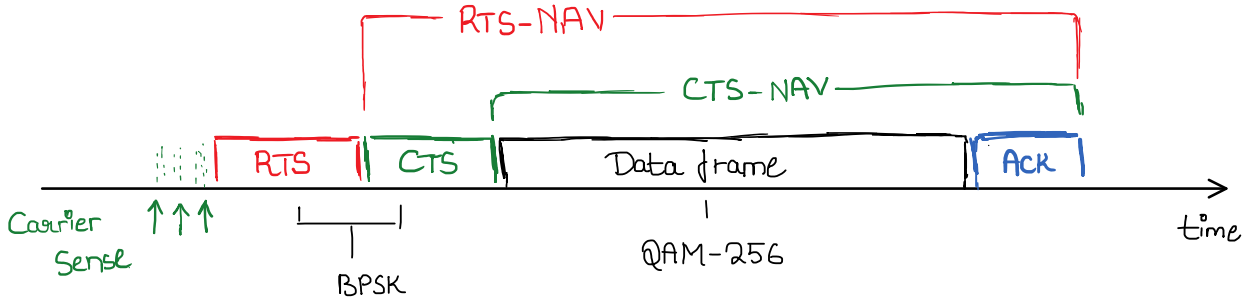
How Long should C stay silent?

Both RTS and CTS have a Network Allocation Vector (NAV) which announce how long the particular node would be busy.

The NAV in RTS and CTS are not the same, see below for reference.

Rule:- Any node hearing the RTS/CTS should be silent for the duration mentioned in the NAV.

RTS → Ack frame clear CTS → Data frame clear



Disadvantages

RTS/CTS should be heard by a large number of nodes. This requires them to be transmitted using BPSK and the data is done using QAM-256. Therefore, even if the size of VCS is small, the time taken to transmit is comparable to the data itself.

- Because of this, RTS+CTS is usually shut-off and only used when needed.

* This protocol is called as CSMA-CA ⇒ Collission Avoidance.

2) Exposed Terminal Problem

Consider the above arrangement where only the adjacent nodes can hear each other. Therefore, there is no problem with such a transmission. However, due to carrier sensing, only one of them will be able to transmit at a given interval.

* Contention Window

A period of time is left after a frame is done being transmitted to allow the receiver to process it. This gap is called as DIFS, Distributed Inter Frame Spacing. Similarly, we need to have a delay in between the different parts of the same frame. That is, a gap between RTS and CTS. This gap is called SIFS - Short Inter Frame Spacing. This is to ensure PHY/MAC processing

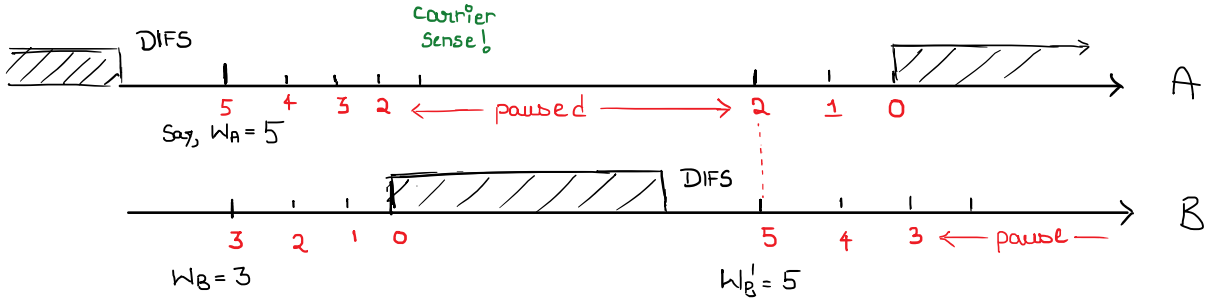
SIFS < DIFS to ensure continuity of frame

- Notice that we donot want all nodes to transmit at once after DIFS because CD takes a long time. To prevent this, we make nodes wait for a random amount of time.

In wifi, time is divided into slots of ~9μs after DIFS. This time should be large enough to accomodate

1) Propogation delay
2) Carrier Sensing + Switching Rx → Tx
3) Division offset

A number W is pulled from the uniform distribution $(0, CW_{max})$. W is decremented by 1 when a period passes without any activity in the medium. The count is NOT reset when a signal is sensed. This is to ensure that all nodes get a chance to transmit.

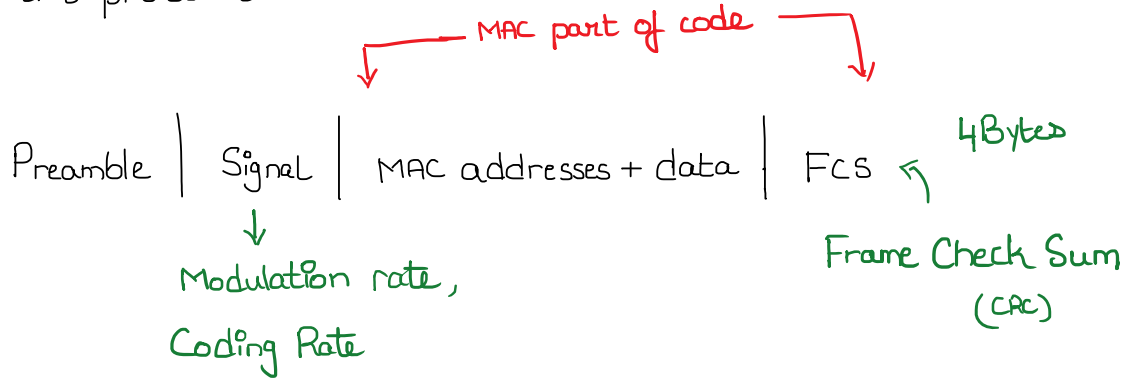


Notice that a collision is still possible if both A, B have the same value of W. The process is repeated after CW_{max} is doubled, similar to Exponential backoff in Ethernet.

$$CW_m = 2 \times CW_m$$

$$CW_m = \min(CW_m, \text{max-allowed-Value})$$

This protocol is called IEEE 802.11. We shall look at the frame structure in this protocol.



Coding Rate – additional bits are added to data to make it resistant to corruption.

$\frac{3}{4}$ coding rate ⇒ out of N, $\frac{3N}{4}$ are data

* Wifi uses 5/6 as its coding rate.

channel width

* IEEE 802.11 g → 64 QAM, 3/4 coding rate, 54 Mbps, 20 MHz

MIMO
- n → 64 QAM, 5/6, 150 Mbps, 40 MHz
- ac → 256, 3/4 5/6, 866 Mbps, upto 160 MHz
- ax → 1024, upto 5/6, 1.2 Gbps, upto 160 MHz

↓ all for SISO

- CSMA-CD and CSMA-CA are decentralized in nature, hence they are especially useful for unlicensed bands, such as wifi. This is because the existence of a central authority is not possible for unlicensed bands.

- Consider a licensed band, such as in the case of 4G. Using CSMA is inefficent in this because it inherently wastes a lot of time for DIFS and SIFS.

* Idea1 – TDMA - Time Division Medium Access

- Divide time into slots, assign two slots to each user - one for Uplink and other for Downlink.

  Uplink — Data sent from user to station - UL

  Downlink - Data sent from station to user - DL

- The entire schedule is called as a Frame and it can be repeated continuously. First few slots in the frame are used inform to DL-MAP and UL-MAP for all users.

- This is efficient as no random waiting time is present, and all time slots are well scheduled without waits (unlike DIFS)

- A guard Band exists so that there is no overlap between signals from different providers.

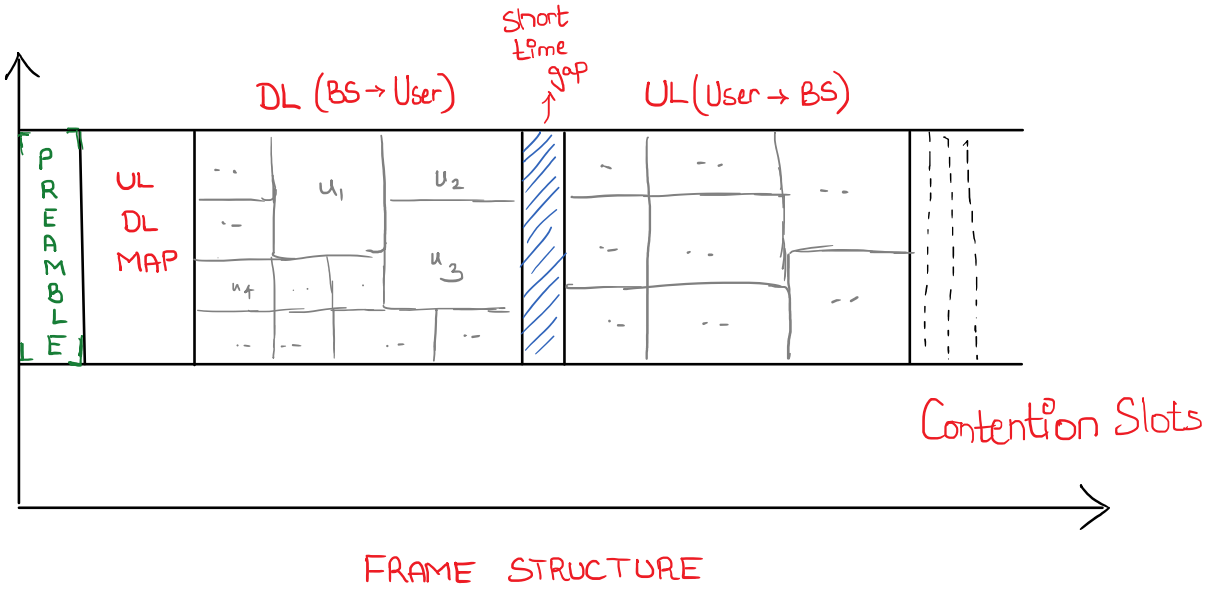
* Idea 2 - FDMA - Frequency Division Multiple Access

- As the name suggests, each user is allocated a frequency band in which it can transmit. Cell phones can isolate frequencies pretty well, so this is viable as well.

* 4G-LTE — OFDMA - Orthogonal Frequency DMA

- This is a combination of FDMA and TDMA.
- The frame starts with a Preamble. This helps in :-

  1) Clock Synchronization
  2) Frame Start
  3) Estimate attenuation
  4) Estimate phase change and time delay



FRAME STRUCTURE

- Each of the u_i in DL and UL are orthogonal, which means that there is no need for guard bands, increasing efficiency.

- Each user would be facing a different degree of attenuation, and this would depend on the frequency as well. After receiving the preamble, each user sends a feedback with the min. attenuation. The maps' slots are generated based on this. Also notice, the map isn't constant

- Contention slots are designated towards hearing the transmission request from the users. Note that this is not pre-determined, meaning that CSMA-CA has to be employed here.
- The information from the contention slots are used to create the next frame's mapping.

* Physical Layer Analysis of OFDM

- The frequency is different for each user, meaning that the analysis can be done for each layer individually.
- Also, this means that different modulation schemes can be used simultaneously.

*

* CDMA – Code Divison Multiple Access – 3G

- Multiple people transmit data in the same time frame and in the same frequency band. This is done by using a Spreading Code.
- CDMA and OFDMA are robust to multipath.

The signals in this case would be of form:

received signal $\leftarrow r(t) = \underbrace{c_1(t)\cos(2\pi f t)}_{s_1} + \underbrace{c_2(t)\cos(2\pi f t)}_{s_2}$

$c_1(t)$, $c_2(t)$ are spreading codes.

- $c_i^2 = 1$
- $\int_0^T c_i c_j \, dt = 0$

Can be generated by randomly varying the values of $c_i(t)$ at -1 and 1

Many users $\Rightarrow r(t) = \sum c_i(t)\cos(2\pi f t)$

Notice that

$$S_1(t) \times C_1(t)\cos(2\pi f t) = A\cos^2(2\pi f t)$$

$$= \frac{A}{2}(1 + \cos 4\pi f t)$$ Remove with low pass filter, as $f \gg 1$

$$S_1(t) \times C_2(t)\cos(2\pi f t) = c_1 c_2 \times A\cos(2\pi f t)$$

$$= c_1 c_2 \times \frac{A}{2}(1 + \cos 4\pi f t)$$

$$= c_1 c_2 \times \frac{A}{2}$$ after passing through low pass filter.

Net signal would look like

$$r(t) \times C_1(t) \cos 2\pi f t = \frac{A}{2}\left(1 + \sum_{j \neq 1} C_1 C_j\right)$$

becomes 0 on integration

$$\int_0^T r(t) \times c_1(t) \cos 2\pi f t \, dt = \frac{AT}{2}$$

Signal obtained!

* L2-Switches

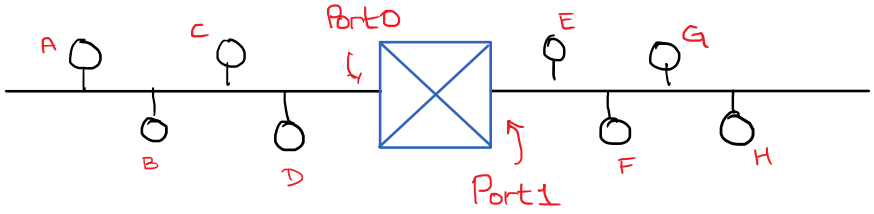
- These are used in the Ethernet, and is done via the MAC addresses.

  Ethernet with a Bus configuration is not scalable due to collisions.

- Switches are used to provide intelligent isolation between parts

  of the network, to make LANS scalable.

- Port is where the switch connects to a LAN.



- The switch transmits signals if the destination is on the other side of the LAN. If the destination and source are on the same side, then it does nothing.

- This is done by storing a Forwarding Table in the switch.

  This table stores the port numbers of all the nodes in the network.

  If receiver port = sender port,

  do nothing

  Else, transmit signal at receiver port.

- Filling up this table manually is not practical. We would like the table to be filled dynamically.

* Filling up Forwarding Table :-

- Each node in the network has a unique MAC address. Similarly, each port has a unique MAC address as well. (MAC - 6 bytes)

- Initially, the switch's table is empty. Every time it receives a frame, it maps the sender to the corresponding port. If the destination's position is unknown, the frame is forwarded to all ports by default.

- Additionally, each entry has an Expiry Time. If the switch doesn't receive a frame within this time, then the entry is deleted. This is to ensure that the table reacts to changes in the topology dynamically.

* Multi-Switch Networks :-

- Loops can be formed when there are many LANs connected without care. This would cause an ∞-Loop when the packet is forwarded by default.

* Spanning Tree Protocol :-

- We would like to disable a few ports to break the Loop, strategically. The steps followed by the protocol are outlined below.

Step1 Elect a switch to be the root switch (Root bridge)

Step2 For each bridge in the network, assign the port closest to the root bridge the root port.

These ports will not be broken finally, and are also called as Active ports.

A tie-breaking tool would be required.

Step 3 For every LAN, choose the port closer to the root and label it as a Designated port.

Step 4 Disable all ports other than the root port or the designated port.

A disabled port can neither listen to signals nor forward signals.

* Electing Root Bridge

- Each bridge has a Bridge ID, and the lowest one is chosen.

Bridge ID = Configurable part (2 bytes) + (MAC address (6 bytes)) ← Smallest MAC chosen by default

- The configurable part is used if a root node is to be forced.

↳ default = 32768 — Reduce to make this the root.

Can be between 0 and 61440

Has to be a multiple of 4096

- Each bridge sends out a signal in the following format.

(Y, d, X)

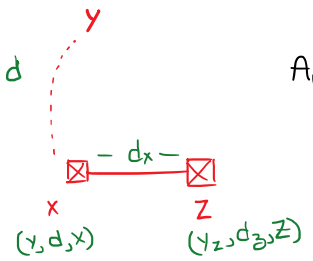
Y: Smallest ID till now

d: dist to Y

X: My ID

- Initially, the signal sent out by all switches would be of the form $(x, 0, x)$.

  As the signal is sent out, each bridge changes the tuple accordingly. It can be seen that all signals would eventually converge at a root.



After Z receives X's signal,

if $Y < Y_z$, then $Y_z \leftarrow Y$ and $d_z \leftarrow d + d_x$ ← found root node

if $Y = Y_z$ but $d_z > (d + d_x)$, then $d_z \leftarrow d + d_x$ ← found shorter path

* Deciding Root port :-