UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Master's Degree in Computer Science

Academic year 2024/2025

WIRELESS NETWORKS FOR MOBILE APPLICATIONS

Prof. Claudio Enrico Palazzi

Written by Michael Amista'

Table of contents

1. Introduction on wireless communication

Modern wireless networks are rapidly expanding in popularity, driven by an increasing number of constantly connected devices and rising demands for higher data rates. This trend has led to significant growth in Wi-Fi technologies and cellular generations (3G, 4G, 5G, and soon 6G) alongside the rise of applications requiring both low and high data throughput.

Wireless technology is now embedded in various interdisciplinary applications:

- **Continuous data modification and upload**: social media platforms, for example, enable users to frequently update and share new content (Web 2.0).
- **Multi-reality experiences**: enabling diverse immersive environments (AR/VR/MR/tele-presence).
- **Ultra responsive to our touch**: enabled by ultra-low latency and high availability (tactile Internet provides instant feedback to touch, enabling real-time, responsive interactions).
- **Real-time connectivity across diverse data demands**: devices communicate and gather data continuously.

Future wireless networks aim to enable seamless, ubiquitous communication between people and devices, facilitating wireless access and cellular connectivity everywhere, often with intelligent data services and network infrastructure. Achieving this vision requires careful attention to constraints like bandwidth, latency, energy consumption, and connectivity.

Designing these networks presents several challenges:

- **Wireless channels are complex and capacity limited**: unlike wired networks, wireless channels face unique constraints: interference, error rates, delay, and nominal speed are key factors to consider.
- **Network planning is challenging**: factors like traffic patterns, user locations, and network conditions are in constant change. Node positioning can vary unpredictably, complicating uniform solutions.
- **Applications impose diverse requirements**: networks must support a variety of applications with unique demands. Some applications are delay-tolerant, while others require immediate responses, which necessitates tailored network management.
- **Energy and latency constraints affect protocol design**: efficient traffic regulation is challenging yet essential on shared channels. Battery constraints are particularly critical in mobile devices, underscoring the need for energy-efficient solutions across network layers.

Multimedia requirements

	Voice	Data	Video	Game
Delay	< 100ms	-	< 100ms	< 100ms
Packet Loss	< 1%	0	< 1%	< 1%
BER	10^{-3}	10^{-6}	10^{-6}	10^{-3}
Data Rate	8-32 Kbps	1-100 Mbps	1-20 Mbps	32-100 Kbps
Traffic	Continuous	Bursty	Continuous	Continuous

Some considerations about these requirements:

- The delay in data transmission is not a crucial factor. In fact, since data is transmitted over several packets, we do not care about the delay of each single packet but instead we care about the time transmission of the whole file, which is the throughput, the total amount of time to transmit a message, not the delay.
- Note there is no packet loss in data transmission, this is due to the way packet transmissions are implemented, allowing to retransmit a packet when it is not received.

From these different requirements, it is obvious that **one-size-fits-all protocols and design do not work well** because each media has specific requirements. Wired networks use this approach with poor results.

In wireless network design, employing a **cross-layer design** is preferable. Unlike traditional designs where layers operate independently, cross-layer design allows for better integration and interaction across hardware, link, access, network, and application layers. This approach is crucial in wireless systems, where constraints such as delay, data rate, and energy efficiency must be balanced. By allowing layers to communicate and share information, cross-layer design improves adaptability, reduces uncertainty, and enhances robustness, ultimately leading to more efficient performance in dynamic wireless environments.

1.1 Current wireless systems

CELLULAR SYSTEMS

A cellular system is divided geographically into cells, each cell uses a different frequency than the nearby cells, but cells that are separated by others can reuse some frequencies. Each cell is controlled by a base station, that manages the transmission and reception of signals between networks and devices.

Note on hand-off: there are means that understand if a handoff is going to happen and duplicate the communication to both antennas in a way to keep the communication stable during the transition (in this way we do not perceive anything during calls).

WIRELESS LOCAL AREA NETWORK (WLAN)

Wireless Local Area Networks (WLANs) connect nearby devices, typically within a range of around 100 meters, allowing them to communicate through an internet access point. Data transmitted over WLANs is broken into packets, and devices share channel access, often using random access methods to send and receive data. While the backbone internet provides a best-effort service, some applications, like video streaming, may experience poor performance due to the shared nature of the wireless channel and the unpredictable delays this can cause. WLANs are a popular choice for local connectivity, providing flexible and convenient access for multiple devices.

Note on the "real" throughput: the nominal Mbps shows the maximum amount of data that our device can receive. The subscription to an operator determines the real amount of data that can effectively be received.

SATELLITE SYSTEMS

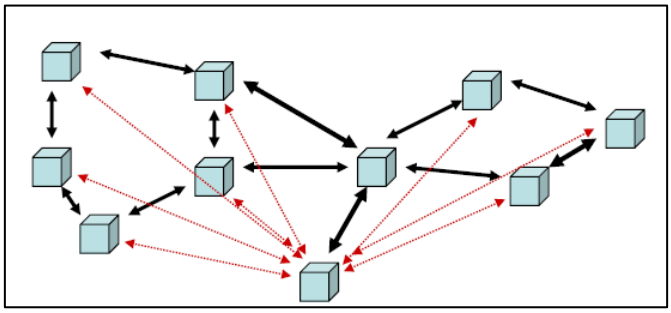
Satellite systems provide coverage over very large geographical areas and operate at different orbit heights, with geostationary satellites (GEOs) at around 39,000 km and low-Earth orbit satellites (LEOs) at around 2,000 km. These systems are typically optimized for one-way transmissions, such as radio and television broadcasting.

BLUETOOTH

Bluetooth is a low-cost, radio frequency (RF) technology primarily designed for replacing cables in short-range communication. Typically, it operates within a range of 10 meters, though it can be extended up to 100 meters through multi hop connections. Bluetooth uses the crowded 2.4 GHz frequency band and supports one data channel with speeds up to 700 Kbps, along with three voice channels. It is widely adopted across industries, including telecommunications, personal computers, and consumer electronics. However, beyond its use as a cable replacement, Bluetooth has relatively few applications, limiting its broader utility.

1.2 Emerging wireless systems

AD-HOC NETWORKS (ANETs)

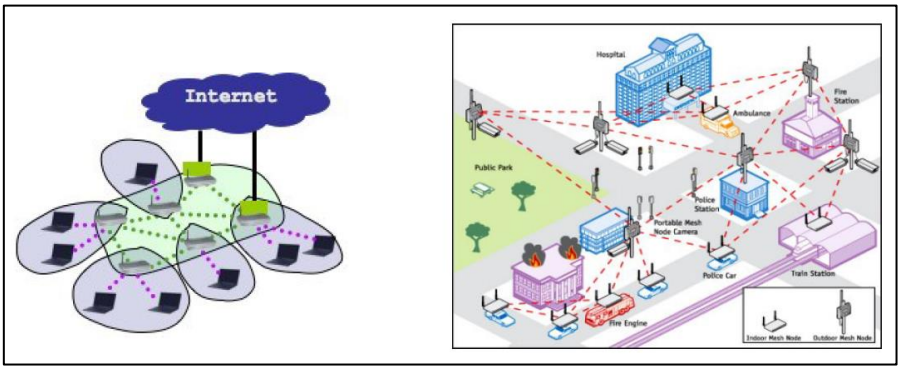


They have a *peer-to-peer* communication architecture, devices communicate with each other without relying on an intermediary (AP or router). Routing can be multi hop to extend area of coverage: to reduce interference devices can pass the data between each other to reach a

further destination. Devices in an ad-hoc network automatically discover and connect with other devices in range, they automatically create routes to reach the other devices (a device when part of an ad-hoc network broadcasts its presence to other devices to build a route). They topology is dynamic, it can change frequently (devices join or leave without requiring manual configuration). In an ad-hoc network there is not a communication over the public internet, the aim is to create a really close network.

It was created in the military field, now they provide a flexible infrastructure for many emerging applications. The capacity of such networks is generally unknown, it depends on how many nodes are part of that network, on the interference, transmission available, how much area they cover. All the strategies (transmission, access, routing) that are used in these networks are ad-hoc. Energy constraints are very important because the devices (nodes) that are part of the ad-hoc network consume battery (usually we are talking about phones or sensor, with vehicles there is not this problem).

MESH NETWORKS



They can be seen as an ad-hoc opportunistic extension of a fixed urban infrastructure. The aim is to create a low cost, high performance wireless coverage. Here is possible to have multiple routers that are connected to different access point wirelessly, this way Wi-Fi coverage can be extended to very large areas without the need to use wired connections between access points. An example could be that there are a few routers (more specifically modems) connected to the internet through cable, and then more routers (mesh nodes) are wireless, and they extend the coverage of the signal in large areas. Unlike traditional networks that rely on a single central router or switch, mesh networks do not have a single point of failure. This allows for improved redundancy and coverage. This means if one node fails or goes offline, data can be rerouted through other nodes.

The main challenges are finding routing protocols to achieve load balancing, QoS, efficient autonomous operations when the infrastructure fails.

SENSOR NETWORKS

This is a specific type of ad-hoc network nodes are sensors used to monitor specific aspects. A sensor can collect data from the environment and send the information back to the central system for analysis and action. Sensor communicates wirelessly. When a sensor collects data and process it, it sends the data to a central node called sink or base station (centralized location) where it can be analysed, this communication can happen through multi hop

communication. Sensors are powered up by batteries or energy harvesting methods, to minimize power consumption. Note that here the driving constraint is the energy.

DISTRIBUTED CONTROL NETWORKS

These are systems designed to control operation across devices and sensors. The nodes that are part of these network perform their specific task, they communicate with each other in a P2P fashion or another way depending on the network architecture. In these systems some distributed control algorithms are implemented to enable the collaboration between nodes. Each node can apply some decision making by performing certain actions.

MOBILE AD-HOC NETWORKS (MANET)

A type of network that establishes wireless communication in dynamic and decentralized environments. In this case the nodes (mobile devices) communicate with each other without the need for centralized control (router or AP). The difference between an ad-hoc network and a MANET is just the emphasis that the nodes are mobile devices (they keep moving), just ad-hoc is a little more static. This kind of networks are mainly created to satisfy a temporary need, are easily deployable and re-configurable.

OPPORTUNISTIC AD-HOC NETWORKS

These are a specific type of ad-hoc networks, nodes communicate opportunistically, they operate in environments where nodes may not always be in direct communication range, so when two nodes come within contact, they can exchange data before they move apart again. Here nodes store data until they encounter another node that can help move that data to a closer destination (store carry forward paradigm). The routing is called opportunistic because data forwarding decision are based on the contacts that occur between nodes, they decide dynamically when to send data.

VEHICULAR AD-HOC NETWORK (VANET)

A specialized form of MANET, designed for communication between vehicles. This kind of network aim to improve road safety, traffic efficiency. The communication must be really fast, it must have low latency.

FLYING AD-HOC NETWORK (FANET)

A specialized form of MANET, designed for communication between aerial vehicles such as drones. Here drones can communicate with each other, and possibly with ground stations or other infrastructures.

UNDERWATER NETWORKS

These networks consist of various interconnected underwater sensors and vehicles deployed in water (oceans, lakes, rivers) to perform tasks like environmental monitoring, disaster prevention. Unlike terrestrial sensor networks, underwater networks face unique challenges due to the properties of water, such as higher communication latency, limited bandwidth, and energy constraints. In this case the medium that transports data is sound (acoustic waves),

radio wave are attenuated in water, that’s why they are used in shallow water. This kind of communication has significant latency, and that is because sound waves propagation in water is a lot slower than electromagnetic waves propagation in the air. Also, the available bandwidth for sound is much narrower compared to the terrestrial wireless communication and that makes difficult to send large amount of data.

RADIO FREQUENCY IDENTIFICATION (RFID)

RFID technology is used to identify and track objects and consists of three main components: RFID tags, readers, and a backend system. RFID tags, which can be attached to objects, contain microchips to store data and an antenna for communication. Tags can be passive (no internal power, activated by the reader's electromagnetic field, and have limited range) or active (powered by their own battery, allowing longer-range communication). RFID readers, either fixed or mobile, emit radio waves to communicate with the tags. When a passive tag comes within range, the reader activates it, and the tag sends its data. Active tags continuously transmit signals. The backend system (a server) processes, stores, and analyses the data received from the readers. RFID operates on different frequency bands, each suited for specific uses, like scanning items in a store instead of barcodes, where tags provide detailed product information.

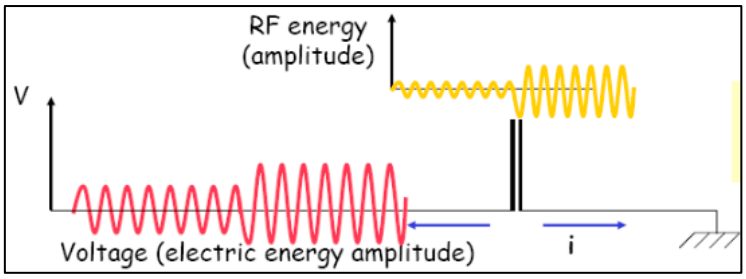
NANO NETWORKS

They refer to communication networks at the nanoscale that consist of nano-sized devices or machines (often referred to as nanomachines) that are able to interact and communicate with each other. Traditional electromagnetic-based communication techniques are often unsuitable for devices at such small scales due to power and size constraints. Instead, communication in nano networks could rely on novel communication paradigms such as molecular communication or terahertz-band communication.

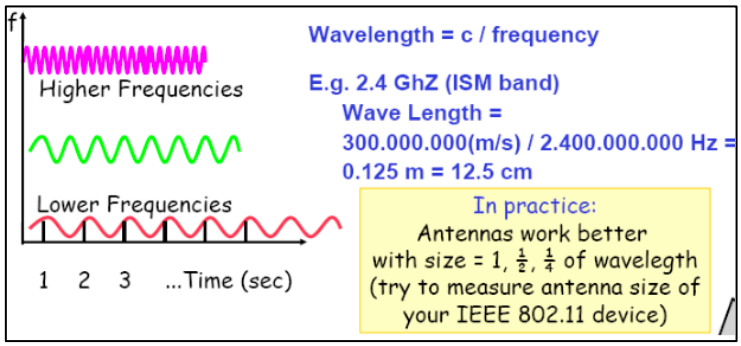
2. Physical layer

2.1 Radio frequencies

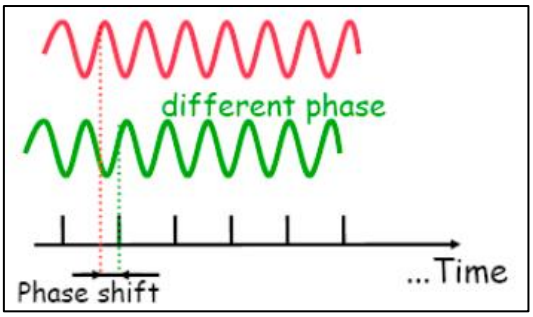
Radio frequencies (RF) refer to the range of electromagnetic waves that are used to transmit and receive information through the air without the need for physical connections like wires or cables. These frequencies range from 3 Hz to 300 GHz, covering everything from very low frequency (VLF) radio waves to extremely high frequency (EHF) microwaves. Electromagnetic energy is generated by high frequency alternate current (AC) in antennas where is converted into RF and vice versa. Radio frequencies are fundamental for wireless communication.



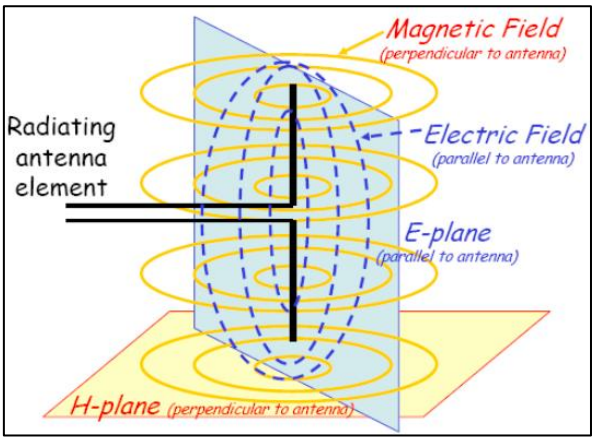
The **amplitude** of the wave (measured as the difference between the highest and lowest peak) is the energy the wave carries, is varied in proportion to the information being transmitted. The greater the amplitude the more energy the waves carry. The higher is the amplitude the further the signal goes. Amplitude depends on the transmission medium (air, outer space, etc.). The transmission power is measured in Watts = Energy / Time = Joule / Sec.



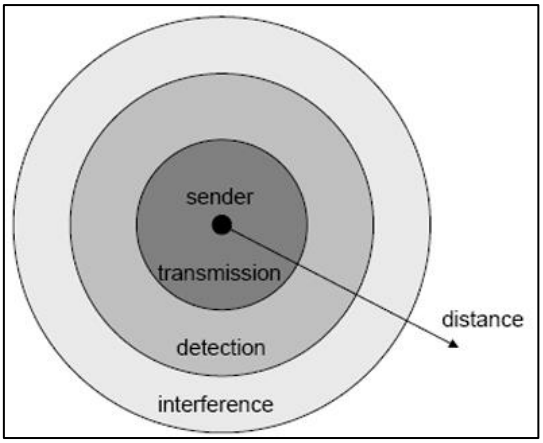
The **frequency** of a wave refers to the number of complete oscillations of the wave that occur in a specific amount of time, typically measured in hertz (Hz). The higher the frequency the more information is transmitted in a specific amount of time but it is difficult to penetrate obstacles. Lower frequencies, instead, have longer wavelengths, can travel longer distances and penetrate objects (penetration can weaken the signal). Frequency allows two nodes to hear communications each other, if both are set on the same frequency value. The frequency is a portion of wireless spectrum assigned to wireless technologies by regional authorities.



The **phase** of a wave refers to the position of a point in time on the wave cycle, indicating the specific stage of the wave's oscillation. It describes the relative displacement between different points of a wave or between two waves of the same frequency. It is the shift of a wave in degrees or radians. The phase can be positive (left-shift) or negative (right-shift).



The **polarization** (physical position of the antenna) refers to the orientation of the electric field in an electromagnetic wave. Polarization can be horizontal if the electric field oscillates parallel to the ground, vertical if the electric field oscillates perpendicular to the ground. Vertical polarization is typically used in WLANs, the antennas must be placed parallel to each other (otherwise less signal is received). RF waves are made of two fields, *electric* that is parallel to the antenna and *magnetic* field that is perpendicular to the antenna.



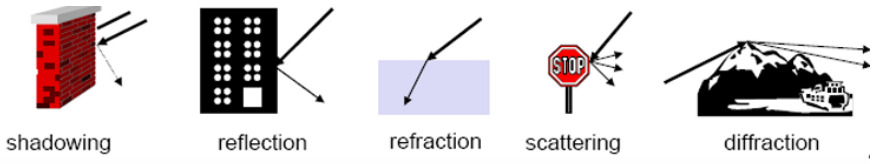
Propagation determines the RF coverage of the signal (how far it can go). After a certain point the signal is no longer detectable, this depends on the strength of the signal itself. Signals become weaker in an exponential decline. Propagation ranges depend on power, obstacles, receiver's sensitivity and many factors. It is quite useful to consider the following ranges of RF detection:

- *Transmission range* = how far the communication reaches. Communication possible, low error rate.
- *Detection range* = how far can the signal be found. Detection of the signal is possible, no communication possible via exchanging messages.
- *Interference range* = the distance at which the signal is too far away from the sender to be detected. Signal may not be detected.

Remember that obstacles can reflect or absorb waves, and it depends on material and frequency used. In general, remember the following rule of thumb:

- **High frequencies = good for short distances/more affected by obstacles**. They get weaker faster at some point the signal cannot be read anymore.
- **Low frequencies = good for long distances/less affected by obstacles**. These signals remain more readable over long distances.

The receiving power is proportional to $1/d^2$ where d = distance between sender and receiver. The receiving power can be influenced by different factors, such as:



Multipath propagation: signals can take many different paths between sender and receiver due to phenomena like reflection, scattering, diffraction.

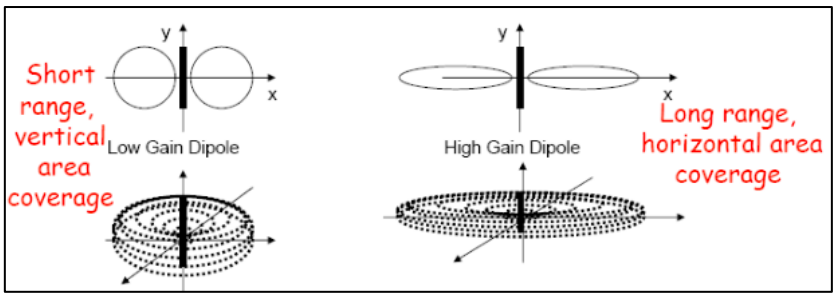
Time dispersion happens when the signal is dispersed over time, it can happen that a signal is scattered into more signals, and some of those annihilate each other. The receiver ends up getting a really bad signal. We use Decibel (dB) as a power measurement to express power loss.

2.2 Antennas

Antennas are devices that transmit and receive electromagnetic waves, enabling communication between devices over a distance, such as radio, TV, cellular, and Wi-Fi systems. They work by converting electrical signals into electromagnetic waves for transmission and vice versa for reception. During transmission, an antenna converts electric signals (which carry information, like voice or data) into electromagnetic waves that can travel through the air. This is achieved by applying an alternating current (AC) signal to the antenna, which generates oscillating electric and magnetic fields.

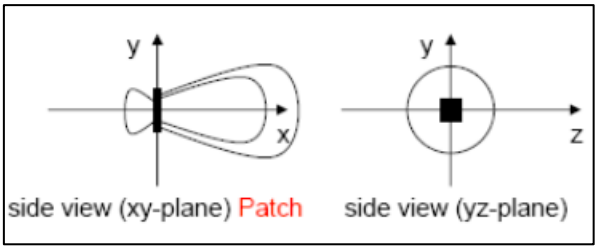
The size of antenna is related to RF frequency of transmissions and receptions, the shape is related to RF radiation pattern. It is important also to notice that antennas are positioned with the aim of reach maximum area coverage.
Real antenna types are: omni-directional, semi-directional, highly-directional.

OMNI-DIRECTIONAL ANTENNA



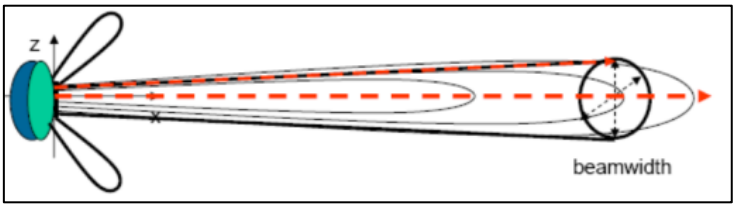
A type of antenna that radiates electromagnetic signals uniformly in all directions around the vertical axis. It is commonly used in situations where coverage in all directions is needed, such as for radio stations, Wi-Fi routers, mobile networks, and some types of broadcast communication, outdoor with point to multipoint connection. An example of this type of antenna is the "dipole antenna"**,** two equal-length conductive elements (usually metal rods or wires) arranged in a straight line. These elements act as the radiating and receiving structure for electromagnetic waves. The dipole antenna has a passive gain due to the shape of radiation. Gain measures how well an antenna can focus energy on a particular direction.

SEMI-DIRECTIONAL ANTENNA

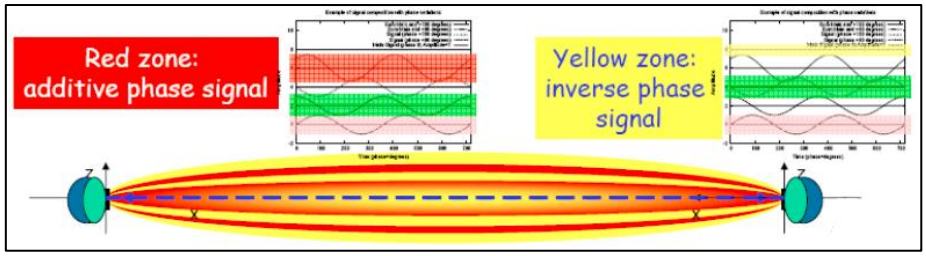


A type of antenna that focuses the radiation of electromagnetic signals in a specific direction (a specific sector), but not as narrowly as highly directional antennas. Semi-directional antennas are often used in applications where moderate coverage over a specific area or in a certain direction is required. Examples are Patch/Panel Antennas that are flat antennas mounted on walls, or Yagi that are rods with tines sticking out.

HIGHLY-DIRECTIONAL ANTENNA

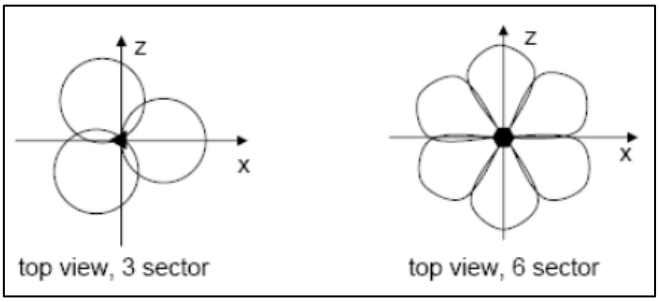


Focus the transmission and reception of radio waves in a very narrow and specific direction. This allows them to achieve greater signal strength and range over long distances compared to omni-directional or semi-directional antennas. These antennas are used in applications where precise point-to-point communication or long-distance signal transmission is required. Examples are the Parabolic dish or the grid.



Line of Sight (LOS) is the unobstructed path (a straight line) between transmitter and receiver. **Fresnel Zone** (FZ) describes the elliptical region around the direct LOS path between a transmitter and receiver. It is essential for understanding how obstacles near the direct path can impact the strength and quality of a radio signal. Most additive RF signal is concatenated in the Fresnel Zone and is important that this zone is free from obstacles. The FZ radius depends on the distance between antennas and the frequency of the signal.
There are some areas of the FZ that add phase to the signal (red ones) and some areas that annihilate the signal (yellow ones).
The beam of an antenna describes the direction and shape of the area where the antenna concentrates its energy. If the FZ is partially obstructed, it is not useful to use higher gain antennas (with smaller degree beam).

SECTORIZED-DIRECTIONAL ANTENNA

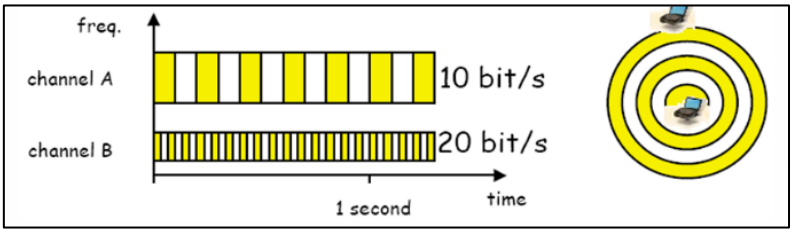


A type of antenna designed to cover a specific, limited area or sector, typically used in cellular networks, Wi-Fi networks, and other communication systems that require focused, directional coverage over a defined geographic region. The use of multiple sectorized-directional antennas allows to divide a wide coverage area into multiple smaller sectors, each served by a different antenna. This enables more efficient use of radio frequency spectrum and improves signal strength within the target areas. They also provide space multiplexing (*channel reuse*: the same frequency channels can be reused in different sectors without significant interference).

2.3 Wireless technologies, coverage and multiplexing

The **wireless spectrum** is the range of all radio frequencies used for wireless communication that may be licensed or unlicensed.

The **bandwidth** of a wireless channel refers to the range of frequencies over which the channel can transmit data, and this range can vary widely depending on the application and environment. Wireless channels can have different bandwidths based on several factors, including the frequency spectrum being used, regulatory limitations, modulation techniques, and the specific technology or communication standard. Basically, **different channels may have different bandwidth because they require less time to accommodate (i.e. to code) one bit on the channel**. Sending bits more frequently means making them more packed (look the figure here below).



There are different wireless technologies that influence the bandwidth:

- **Narrowband radio system** where transmission and reception are executed using a single radio frequency. Undesired cross talk between channels requires coordination and license for each site. There is low data rate.
- **Spread Spectrum technology**, used to spread a signal over a wide range of frequencies, making it less susceptible to interference, jamming, and unauthorized interception. There are two main types of spread spectrum techniques:
    - **Frequency Hopping Spread Spectrum (FHSS)**, where the signal does not stay on one frequency for long but moves across several frequencies within the band in a synchronized way known only to the transmitter and receiver. To unintended receiver FHSS appears as impulse noise.
    - **Direct Sequence Spread Spectrum (DSSS)**, the signal is spread across a wider frequency band by multiplying the original data signal with a high-rate pseudo-random noise (PN) sequence, also called a chipping sequence. Each bit of data is transmitted as multiple bits called "chips," which effectively spreads the signal. To unintended receiver DSSS appears as low power Wideband noise.
- **Infrared technology** is LOS or diffused, short range. The frequencies are just below the visible light, cannot penetrate opaque objects, it allows high data-rate potential.

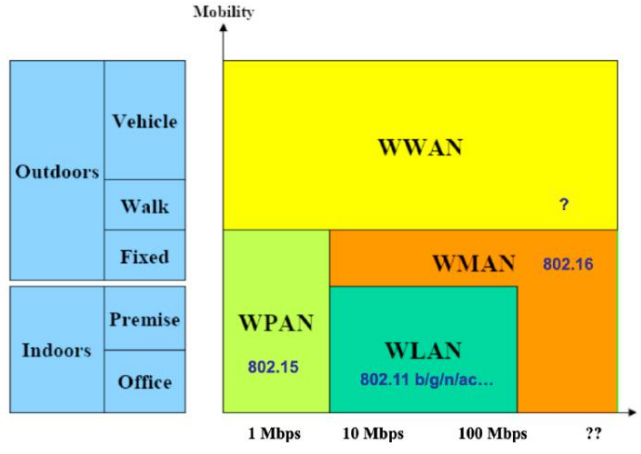
Comparison:

	PROS	CONS
Frequency Hopping Spread Spectrum (FHSS)	• Use less power than DSSS • Lower cost • Increased security due to frequency switching	• Lower throughput than DSSS
Direct Sequence Spread Spectrum (DSSS)	• High performance • Low interference • Increased security due to chip coding	• Expensive
Narrowband Microwave	• Long distance	• Line-of-sight with satellite dish • Requires FCC license • Not designed for WLAN use
Infrared	• High bandwidth	• Easily obstructed • Inexpensive

Radio transmission coverage refers to the geographic area or range over which a radio signal, transmitted from a source such as a transmitter, antenna, or base station, can be effectively received. In wireless communication systems, the coverage area is the region where the transmitted signal remains strong enough for reliable communication between the transmitter and receiver. With Tx we indicate the transmission power with which the radio transmitter sends out a signal. *Symmetric links* are communication channels where the data transfer rates are equal in both directions. *Asymmetric links* are communication channels where the data transfer rates differ between the two directions.

Wireless Network Coverage

- Wireless Wide Area Network (WWAN)
  - geographic coverage (e.g. satellite, cellular)
- Wireless Metropolitan Area Net. (WMAN)
  - Metropolitan coverage (e.g. town, large campus)
- Wireless Local Area Network (WLAN)
  - local area coverage (e.g. campus, building, home)
- Wireless Personal Area Network (WPAN)
  - reduced local area coverage (e.g. house, office)
- Wireless Indoor Area Network (indoor)
  - short range coverage (e.g. room, office)



Multiplexing is a technique used to combine multiple signals or data streams into a single transmission channel or medium. The primary goal of multiplexing is to make efficient use of available bandwidth by allowing multiple communications to occur simultaneously over the same physical medium, such as a wire, fiber-optic cable, or radio frequency spectrum. By using multiplexing, multiple signals can share the same channel without interfering with each other, which improves the capacity and efficiency of communication systems.

There can be different types of multiplexing:

- **Frequency Multiplexing**. The available bandwidth of the channel is divided into multiple frequency bands. A channel gets a certain band of the spectrum for the whole time. No dynamic coordination is required, it works for similar systems, it is not flexible and there is waste of bandwidth if the traffic is distributed unevenly. Each signal is transmitted on a different frequency, allowing several signals to be sent simultaneously without interference. It is used in radio and tv broadcasting.
- **Time Multiplexing**. Multiple data streams share a single channel by dividing the available time into separate time slots. Each signal is assigned a specific time slot, during which it can transmit its data. It is used in digital telephone networks, where multiple voice calls are combined over a single line. Precise synchronization is necessary, only one carrier ate the time in the medium.
- **Code Multiplexing**. Each signal is assigned a unique code. All signals (all channels) are transmitted simultaneously over the same frequency spectrum, but because each signal has a different code, the receiver can distinguish between them. It is used in mobile communication systems like 3G and GPS. Coordination and synchronization are not necessary, has a good protection against interference, has lower data rates and more complex signal generation. It is implemented using spread spectrum technology.
- **Space Multiplexing**. Multiple signals are transmitted simultaneously using separate physical paths or spatial channels. Each signal is assigned a distinct physical location, such as different cables, antennas, or beams, preventing interference. It allows for high data transmission rates and efficient utilization of spatial resources but may require additional hardware and infrastructure.

3. MAC layer

Medium Access Control (MAC) layer is primary responsible to manage how devices in a network gain access to a shared communication medium to avoid collisions, such as a wireless channel or a wired Ethernet network and transmit data over it.
Goals: low latency, good channel utilization, best effort + real time support.

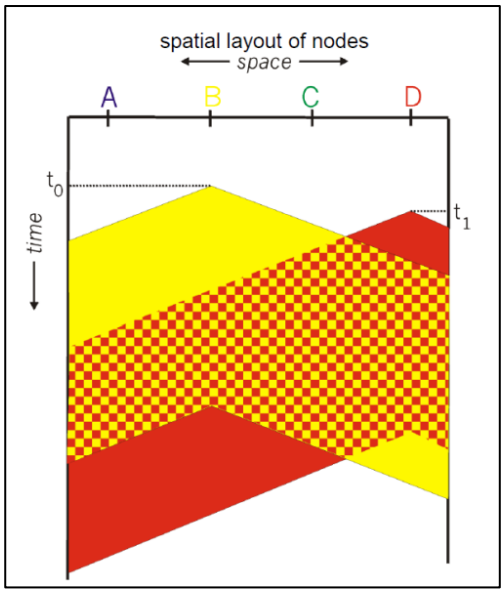
As in a human conversation…
- Everybody should have the chance to talk
- Do not speak until it is your turn
- Do not monopolize the conversation
- Raise your hand if you have to ask for something
- Do not interrupt while somebody is talking
- Do not fall asleep while somebody is talking

Channel Access Problem arises in communication networks when multiple devices or nodes need to share a common transmission medium (such as a wireless channel or wired bus) to send data. The challenge is to ensure that devices can transmit data without interfering with each other, which could cause data collisions, leading to communication failures, delays, or packet loss. Simultaneous communication is not possible, for this reason there are several MAC protocols that schedule communication, maximizing the number of communications while ensuring fairness among all transmitters.

The simplest idea of "transmit and pray", where every node sends data hoping that other nodes do not transmit at the same time, is wrong because it results in a lot of collisions.

One of the protocols that is used to solve this problem is **Carrier Sense Multiple Access (CSMA)**, where the sender listens to the channel before sending data, it can be of two types: CSMA/**CD (Collision Detection)** for wired Ethernet: devices "listen" for an idle channel and back off in case of a collision. CSMA/**CA (Collision Avoidance)** for Wi-Fi: devices try to avoid collisions before they occur by using techniques like random backoff timers.

Collisions can still occur despite its mechanism to "sense" the channel before transmission. This is because CSMA only reduces the chances of collisions but does not eliminate them entirely. The main problem that can lead to collision is the **propagation delay** which refers to the time it takes for a signal to travel from one device to another over the communication medium. Even if a device senses the channel as idle and begins transmitting, another device located far away might also start transmitting because it has not yet "heard" the ongoing transmission. This can lead to a collision, particularly in networks with larger distances between devices.



Consider the following example:

At time t_0, node B initiates a transmission to node C. Observing the time axis, we can see that this transmission is not instantaneous; it takes a certain amount of time to propagate. At a later moment t_1, node D also attempts to transmit. D first checks the channel, and since it appears to be free at that instant, it begins its transmission. However, due to the non-zero propagation delay of the initial transmission between B and C, D's transmission collides with the ongoing propagation, leading to a collision.

- Considering *B* = available bandwidth on the channel

- Desirable features:
  - *Efficiency* in bandwidth use (sum of rates = *B*)
    - If just one node, it should transmit at *B* rate
  - *Resilience*: Avoid collisions
  - *Fairness*: If *M* nodes want to transmit each should have *B/M* bandwidth available (in average)
  - *Robustness*: the protocol should be decentralized (no single point of failure)
  - *Simplicity*: the protocol should be easily implementable

MAC PROTOCOL APPROACHES

Controlled Access MAC Protocols, where devices must coordinate with each other (or a central authority) to access the shared medium. Only one device can transmit at a time, and a predetermined control mechanism ensures no collisions. They can be divided as:

- Centralized: there is an entity that is responsible to regulate the access to the channel (FDMA, TDMA, CDMA).
- Distributed: the access to the channel is controlled by a distributed application, with peer nodes (token ring).

Random Access MAC Protocols, where devices transmit data whenever they have it at full channel data rate R, without a predetermined schedule or control mechanism. These protocols do not require any centralized control over the access to the shared medium. If the transmissions of two or more nodes collide, they retransmit at random times. The protocols specify how to detect collisions and how to recover from them. They can be divided into:

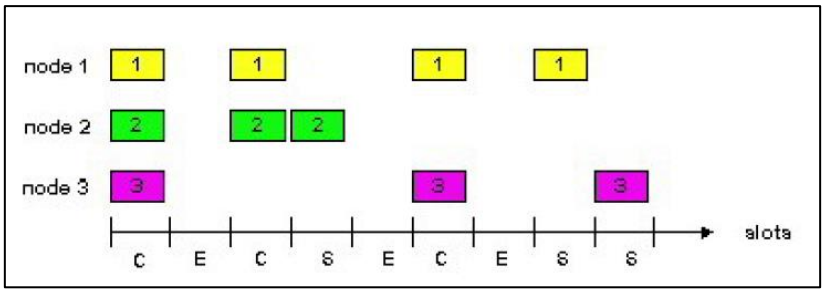
- Without Carrier Sense (ALOHA, Slotted ALOHA).
- With Carrier sense (CSMA, CSMA/CD, MACAW).

3.1 ALOHA protocols

ALOHA is a MAC protocol developed in the 70's by University of Hawaii to have islands able to communicate to each other. Two different versions: Slotted ALOHA and Pure ALOHA.

SLOTTED ALOHA

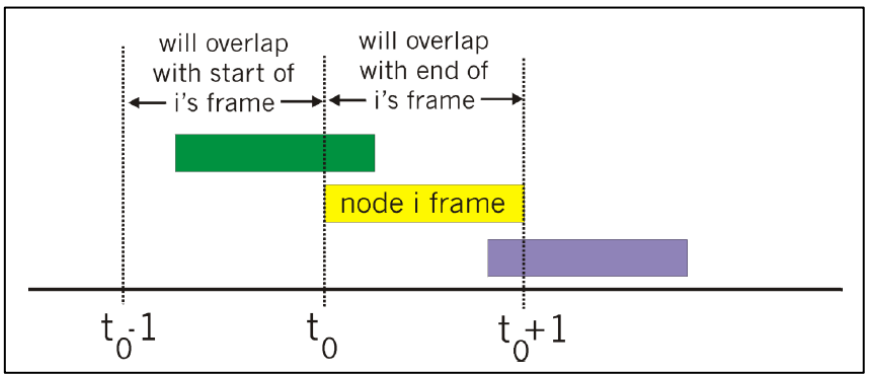
Time is divided into equal slots, and devices can only transmit at the start of a time slot. This reduces the probability of collisions compared to pure ALOHA. If a collision occurs the source retransmits the packet at each slot with probability p (possibility of empty slots), until successful. It is fully decentralized, in fact there is not any central authority. The throughput efficiency is 1/e, meaning that we are using less than half of available bandwidth.

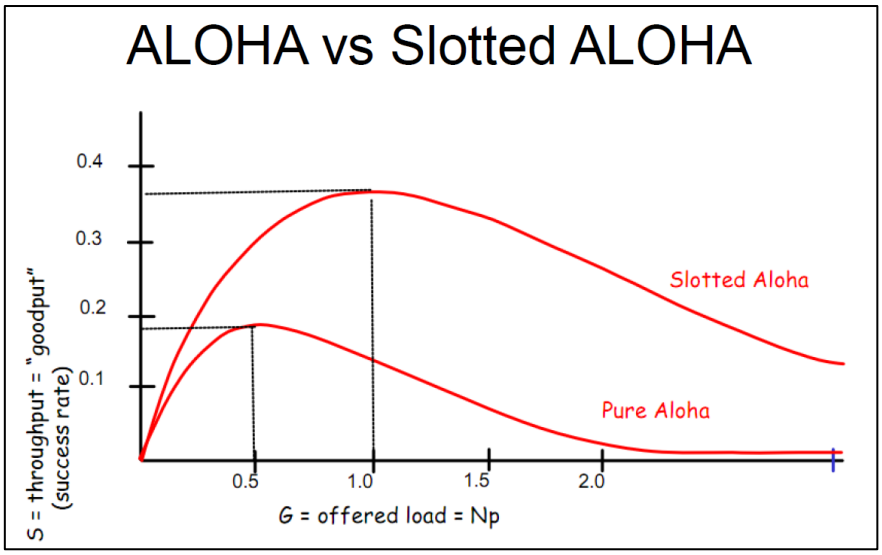


Note: the main idea is to minimize as much as possible the number of empty slots, being more aggressive in transmissions, otherwise precious bandwidth is wasted. Nodes should be synchronized with time slots, otherwise there will be collisions, and this leads to more complexity to manage.

PURE ALOHA

This is a simpler version where devices transmit without checking whether the channel is busy and without waiting for the beginning of a slot (in fact here slots are not used). Collisions are detected afterward, and retransmissions are attempted. The throughput efficiency is worse than unslotted ALOHA: 1/2e (since there is no synchronization and looking the picture each transmission can occupy on average 2 slots, the current and next transmission one).





3.2 CSMA protocols

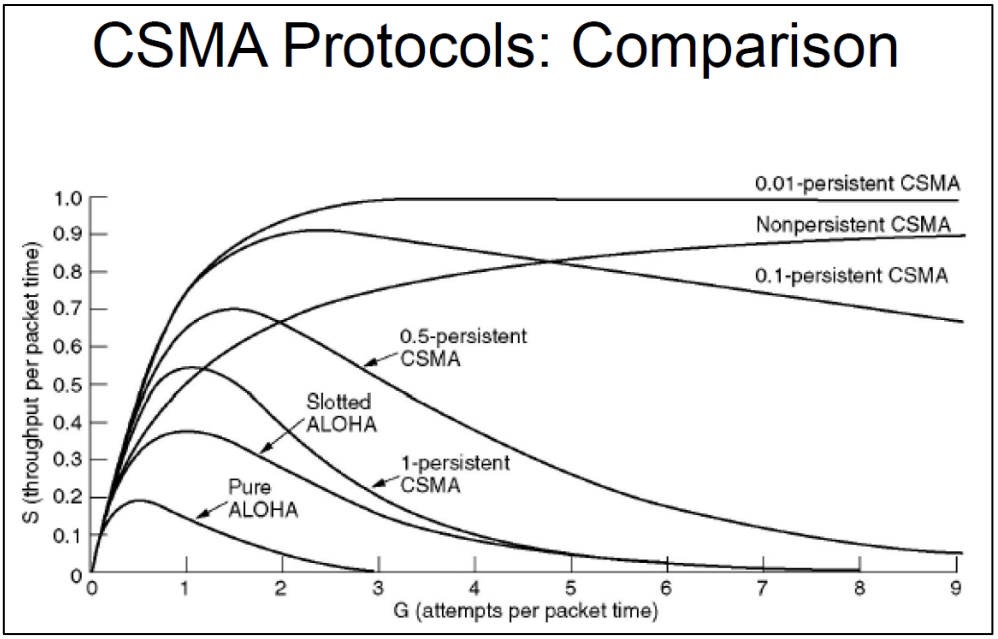
Low performances of Pure Aloha and Slotted Aloha are due to the lack of coordination among nodes, efficiency can be improved if each node behaves coherently with what other nodes do. In carrier sense protocols each node keeps listening to the channel to be aware of what other nodes are doing.

CSMA (Carrier Sense Multiple Access)

CSMA is designed to reduce collisions by requiring each device to first "listen" to the medium before transmitting. This "carrier sensing" ensures that the medium is free (i.e., no other device is transmitting) before starting a transmission, which helps minimize the risk of two devices transmitting at the same time. There are different versions of CSMA:

- **1-persistent CSMA**. "1" refers to certainty. The station will definitely transmit as soon as it detects the channel is idle, with a probability of 1 (100% certainty). The term "1-persistent" refers to how aggressively the station checks the medium and attempts transmission once it becomes idle, in this case the state of the channel is checked continuously. The station first senses the medium to determine if it is busy or idle. If the channel is busy, the station continues to monitor the channel continuously (i.e., it "persists" in checking the medium). If the channel is idle, the station immediately transmits its data packet. If two or more stations sense the channel as idle at the same time, they will both (or all) transmit immediately, leading to a collision. If there is a collision the transmitting node/s waits for a random time and then try to retransmit again (a sort of “desynchronization” between nodes that transmit simultaneously).
- **nonpersistent CSMA**. Unlike the more aggressive 1-persistent CSMA, non-persistent CSMA takes a more conservative approach by introducing a random delay before attempting to transmit after detecting that the medium is busy. Because stations do not continuously sense the channel when it is busy, collisions are less likely in non-persistent CSMA than in 1-persistent CSMA.

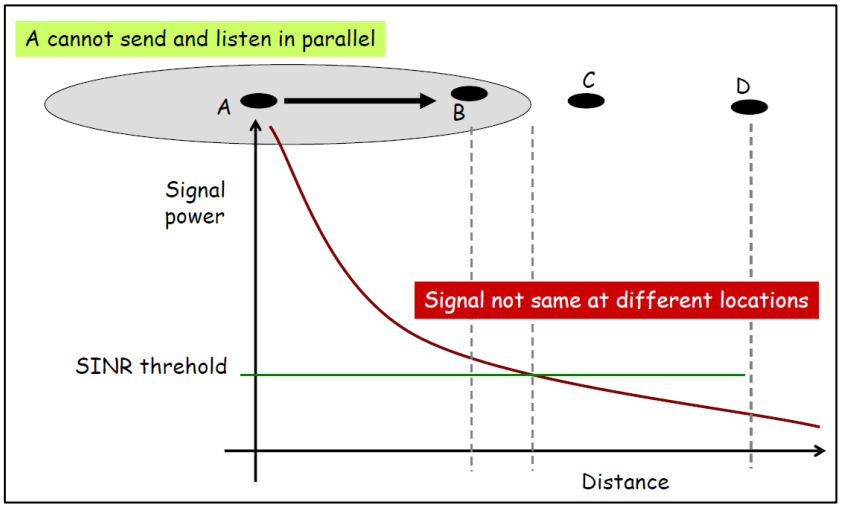
- **p-persistent CSMA**. It is a slow based system, which means the network operates by dividing time into discrete slots. Stations can only transmit at the start of these slots. If the medium is idle, the station transmits the data with a certain probability p during the next available time slot. If the station does not transmit (with probability 1−p), it waits for a random time and tries again the procedure. This process repeats until the station either transmits the packet or the channel becomes busy due to another station's transmission. The probability p represents the likelihood that a station will transmit as soon as the medium is found idle. When p is high (close to 1), the station will transmit almost immediately, like 1-persistent CSMA.



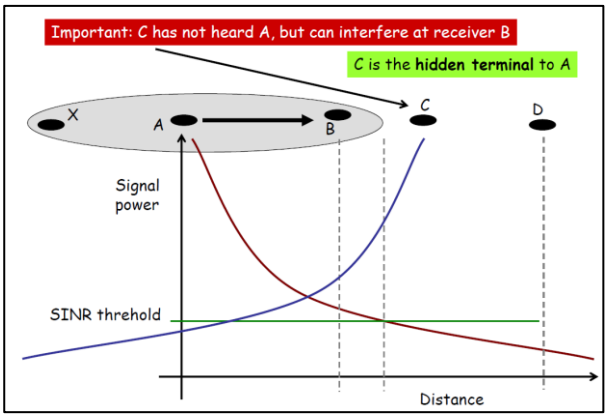
CSMA/CD enhances traditional CSMA by enabling devices to detect collisions within a few bits of transmission, allowing them to abort transmission immediately and minimize channel wastage. This approach is typically implemented with persistent transmission, where devices continuously monitor the channel and resume transmission once it is free. In wired LANs, such as Ethernet, CSMA/CD can achieve near-optimal channel utilization (approaching 1) when the propagation delay is small relative to packet transmission time. Collision detection **in wired LANs is straightforward**, as devices can monitor signal strength, detect code violations, or compare transmitted and received signals.

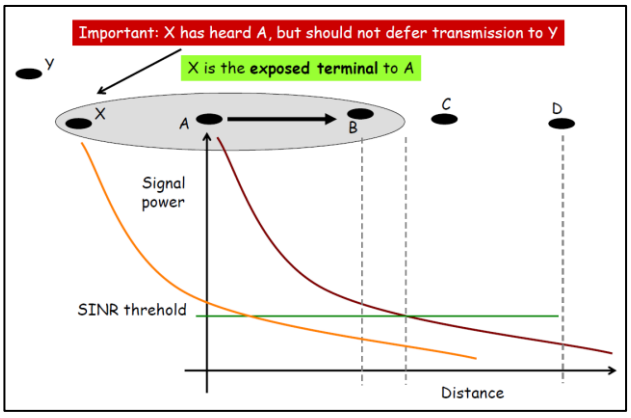
However, CSMA/CD is **impractical in wireless LANs**. When a device transmits a signal, it generates a significant amount of power to send data through the air. To avoid interference or potential damage from this powerful outgoing signal, the receiver circuit of the device is usually turned off during transmission. This means that, while the device is sending data, it cannot "listen" to the channel to detect if another device is also transmitting at the same time.

3.3 Hidden/Exposed terminal problem



The SINR threshold represents the bound in which the signal cannot be detected anymore. So, when A is transmitting something, C (that is out of A's range) cannot detect the signal sent by A since it is interpreted as "background noise", being under SINR threshold. Once C senses the channel and determines is idle (because it cannot hear A's transmission), C tries to transmit something to B, resulting so in a collision with A's signal.





In the **hidden terminal problem**, a node (C) that cannot detect a sender's (A) signal may still interfere with its receiver (B), causing collisions.

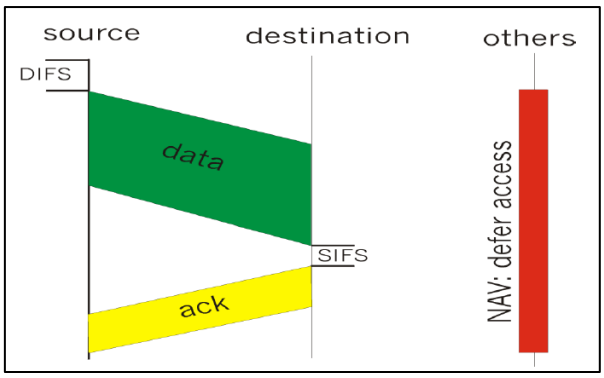
In the **exposed terminal problem**, a node (X) hears another node (A) transmitting, but mistakenly assumes it must remain silent, even though its transmission to another receiver (Y) would not interfere.

3.4 802.11 protocol

The **Multiple Access with Collision Avoidance (MACA)** and **MACA for Wireless (MACAW)** protocols emerged as improvements over the traditional CSMA/CD and CSMA/CA methods, particularly for wireless networks. Both protocols were developed to address key limitations in wireless communication, such as the hidden terminal and exposed terminal problems, and to enhance performance in environments where collision detection is not feasible due to the nature of wireless channels.
MACA introduced a collision avoidance mechanism that does not rely on sensing the channel during transmission (like CSMA/CD) but instead uses a handshake method based on short control messages. In response to the limitations of MACA, MACAW (MACA for Wireless) was introduced by V. Bharghavan et al. in 1994 as an enhancement to MACA, with a focus on improving fairness, throughput, and reliability in wireless communication.
These led to **IEEE 802.11 standard** that defines MAC protocol; unlicensed frequency spectrum bands: 900MHz, 2.4GHz, 5GHz.

The 802.11 standard regards in particular Wireless LANs (WLANs):

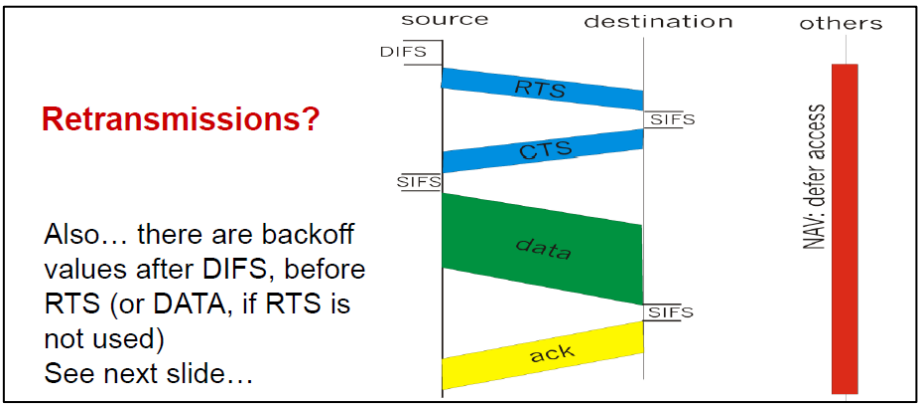
- Mostly indoor.
- Base station (like cellular); or ad hoc networking (mostly point to point).
- Standards: IEEE 802.11 (various versions); HyperLAN(ETSI); Bluetooth.



CSMA version of the protocol

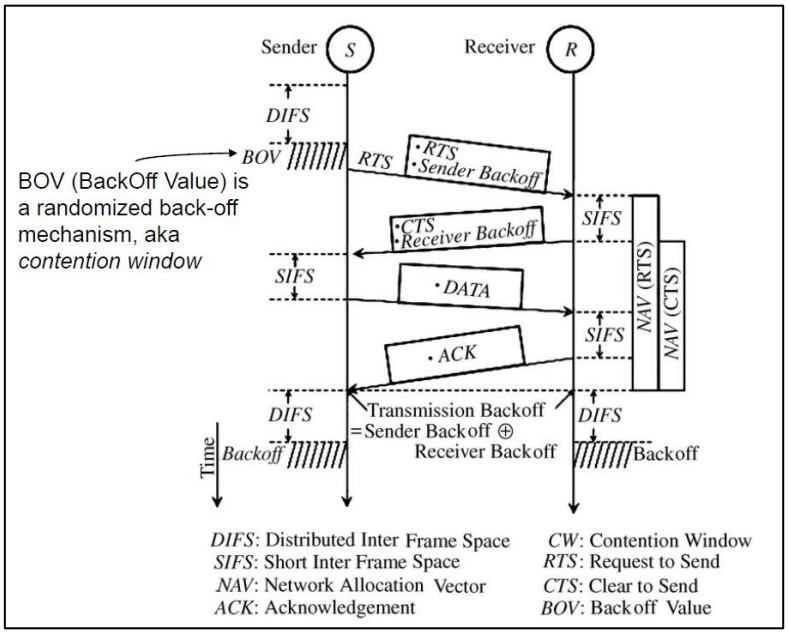
Where the source senses the channel to see if it is idle. If the channel has been idle for a duration equal to **DIFS** seconds, the station proceeds to transmit its data. After successfully receiving a frame, the receiving station sends an **ACK** back to the sender. This occurs after a shorter inter-frame space known as **SIFS** (shortest waiting time between two consecutive transmissions). The ACK ensures that the sender knows the data was successfully received, which helps avoid uncertainty due to potential collisions or packet loss. If the channel is sensed as busy (another station is transmitting), the station does not immediately try to send the frame again. Instead, it enters a **binary exponential backoff** process, which means it waits for a random period before checking the channel again.
The **NAV** is a timer set by each station to represent the amount of time the channel is expected to be busy. The sender declares the duration of its transmission (the time needed to send the

frame and receive the ACK) in the frame's header. This value is used by other stations to set their NAV timers, preventing them from transmitting until the declared time has passed.
802.11 CSMA is inefficient in presence of hidden terminal, the solution is using CSMA/CA.

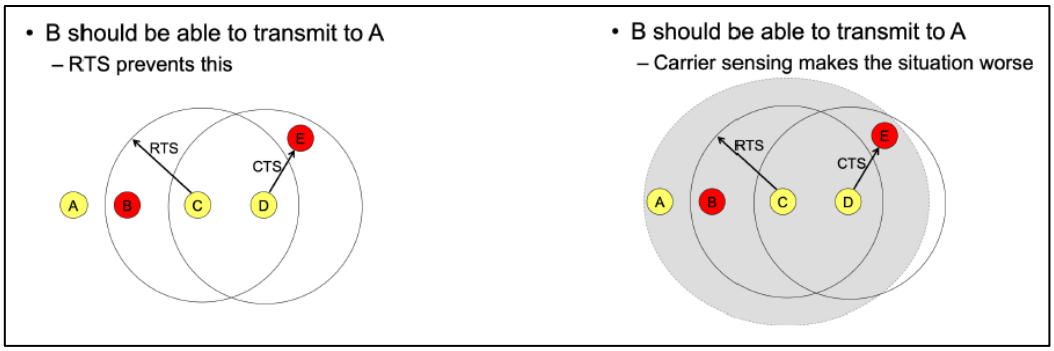


CSMA/CA version of the protocol

In this case we consider CSMA/CA with **RTS/CTS (Request to Send / Clear to Send)**. RTS and CTS frames are used to freeze stations near the transmitter and the receiver, so they do not sent data while the two stations communicate. When a station sends an RTS, any station within the range of the transmitter (but not involved in the communication) hears this RTS and knows that the transmitter is about to engage in communication. These stations will "freeze" their transmissions. Stations near the receiver (which may not have heard the RTS because they are hidden from the transmitter) hear the CTS. This "freezes" them as well.



The above figure describes the actual procedure of sending data, considering also backoff values after DIFS and before RTS. In CSMA/CA, after a station senses the channel is idle for DIFS, it does not transmit immediately but instead enters the backoff phase (BOV), where it selects a random wait time before transmitting. Anyway, with CSMA/CA **the exposed terminal problem still occurs**.



To RTS/CTS or Not to RTS/CTS?

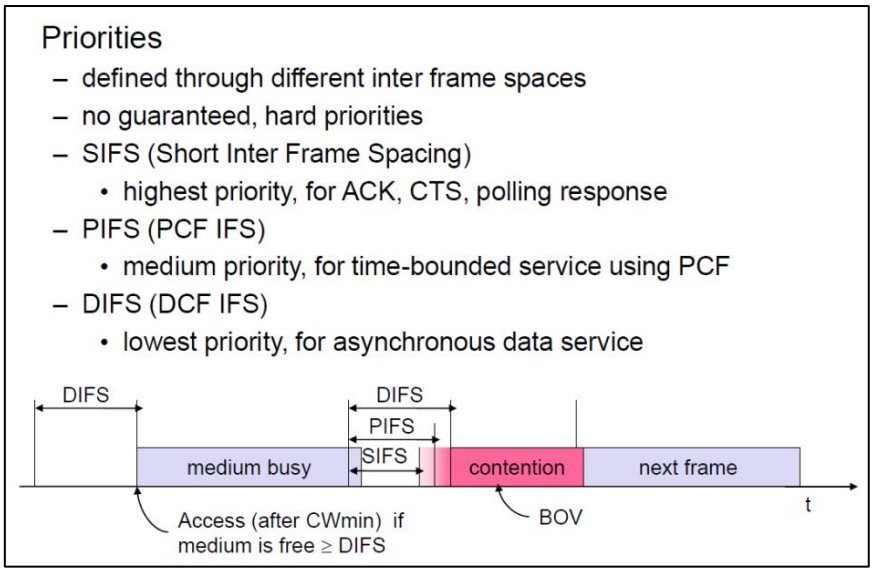
- 802.11 does addresses the hidden terminal problem to RTS/CTS
- Two simultaneous RTS messages sent by two different nodes may result in a collision; so no improvement with respect sending directly the two data messages?
  - Actually RTS messages are much smaller (few bytes) than a data message. So the probability of a collision is smaller (even if not zero). This is way they are an improvement at the cost of the limited overhead of their transmission.
  - If data messages sent are very small (e.g., VoIP or gaming messages), then there is no point in using RTS/CTS and actually they slow down (a little bit) the transmission of data messages and represent an overhead that, although little, is comparable to the amount of traffic generate by the application.

Main idea: **use RTS/CTS with large data transmissions** to avoid collisions and improve overall efficiency, especially in networks prone to the hidden terminal problem. **Avoid RTS/CTS for small packets** (e.g., VoIP, gaming), as it adds unnecessary overhead and can slow down communication.

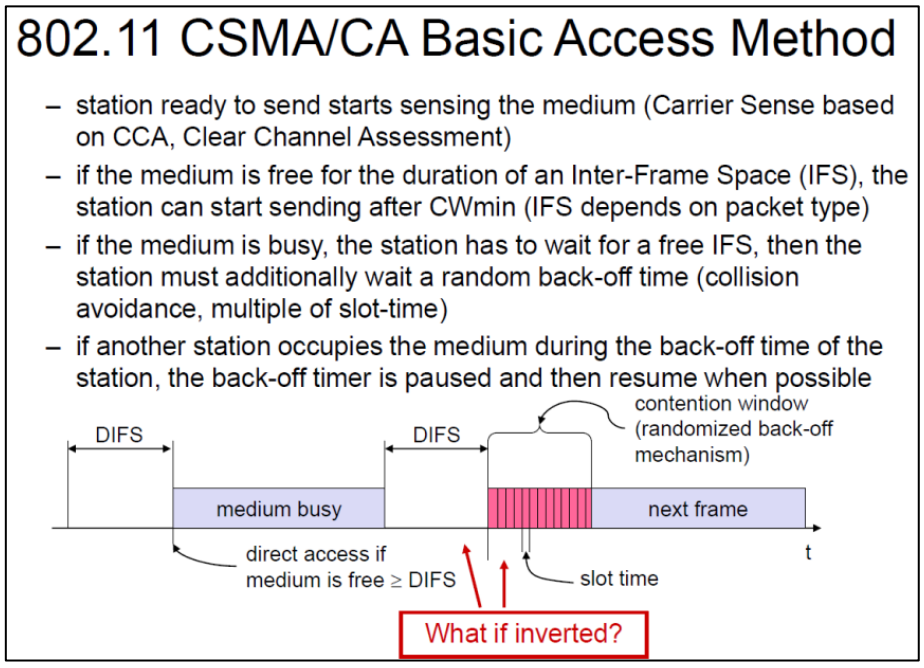
802.11 does not solve HT/ET problems completely, it only alleviates the problem through RTS/CTS and recommends larger CS zones. The search for the best MAC protocol is still on.

IEEE 802.11 MAC Layer

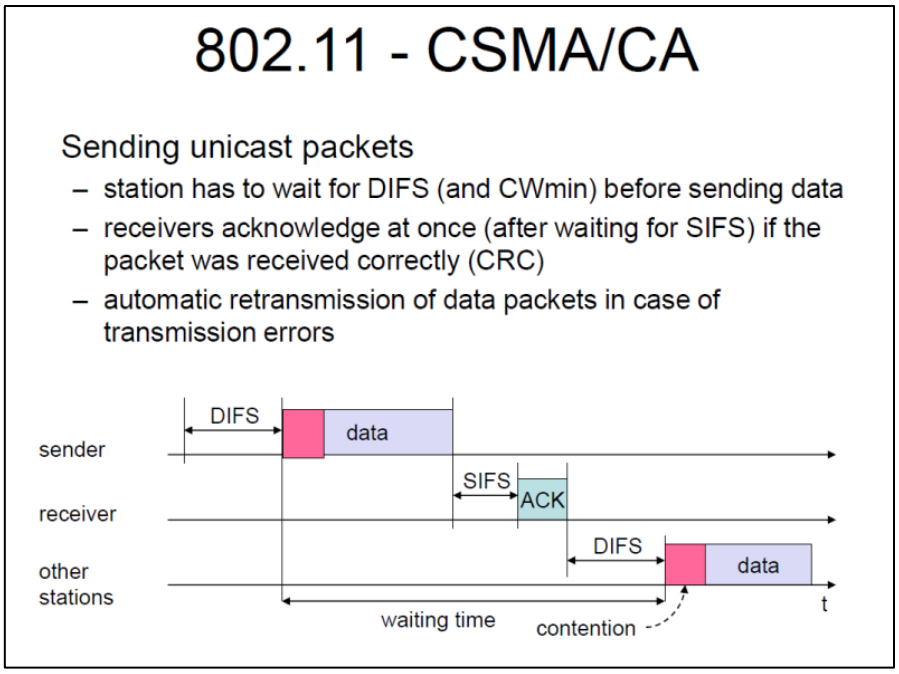
- Access methods:
  - MAC-DCF CSMA/CA (mandatory)
    - collision avoidance via randomized back-off mechanism
    - minimum distance between consecutive packets
    - **ACK** packet for acknowledgements (not for broadcasts)
  - MAC-DCF w/ RTS/CTS (optional)
    - Distributed Foundation Wireless MAC
    - avoids hidden terminal problem
  - MAC- PCF (optional)
    - access point polls terminals according to a list



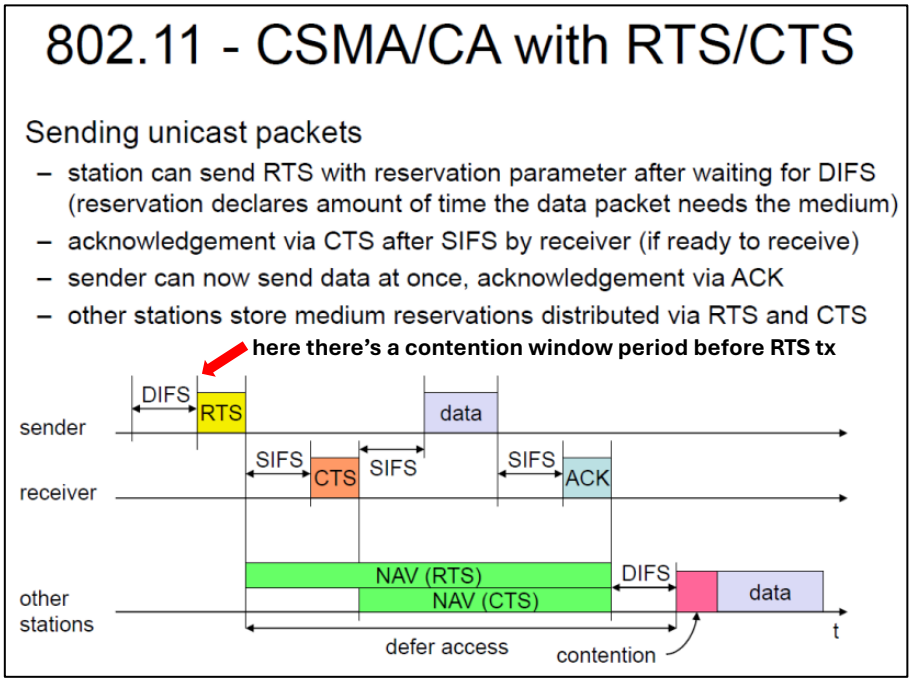
Priorities are managed using different **Inter-Frame Spaces (IFS)**. These are periods of time that stations must wait before they are allowed to transmit data on the wireless medium. The idea is to prioritize certain types of traffic by giving them shorter waiting times, allowing them to access the medium more quickly than others.



Inverting the contention window (backoff timer) with the DIFS duration would disrupt the CSMA/CA mechanism, leading to increased collisions, unfair access, and reduced efficiency. Currently, DIFS ensures a short wait time before accessing the channel, while the randomized backoff prevents simultaneous transmissions. If the backoff occurred first, stations would delay their transmissions unnecessarily even when the medium is idle, reducing throughput. Additionally, multiple stations might finish their backoff at the same time, increasing the likelihood of collisions. This inversion would also create unfair access, as stations with lower backoff values would always transmit first, potentially starving others.



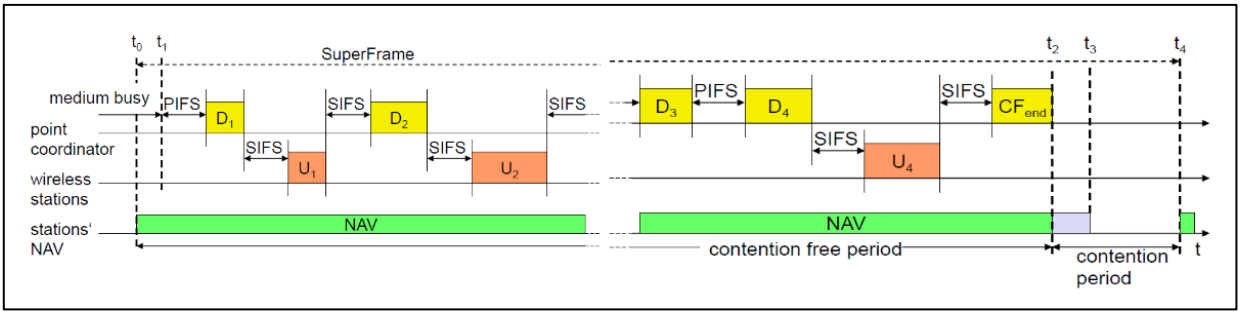
The receiver successfully receives the packet and verifies its integrity using CRC (Cyclic Redundancy Check). If the sender does not receive an ACK within a certain time, it assumes that the packet was lost (due to a collision or transmission error) and will automatically retransmit the packet after waiting for another backoff period.



In this case we are analysing the CSMA/CA with RTS/CTS. Here an additional handshake mechanism is introduced to further reduce the chances of collisions, particularly due to the hidden terminal problem. After waiting for DIFS and a random backoff time, instead of sending the data directly, the station sends a RTS frame to the receiver. The RTS frame includes a reservation parameter (NAV), which tells all other stations how long the medium will be needed

for the data transmission. If the receiver is ready to accept the data, it responds with a CTS frame. The CTS frame also includes the same reservation parameter as the RTS, informing all stations within range of the receiver to refrain from transmitting for the reserved duration.

MAC-PCF (Point Coordination Function) like polling



MAC-PCF is an optional mode in the IEEE 802.11, that provides a centralized polling mechanism for controlling access to the wireless medium. In PCF mode, the Access Point (AP) takes control of the wireless medium for a portion of the time, known as the **Contention-Free Period (CFP)**. During this time, the AP acts as a Point Coordinator (PC) and polls the stations (nodes) to check if they have data to send. The **stations do not compete for access**; instead, they are polled in turn by the AP. The AP sends a poll frame to a station, which grants that station **exclusive access to the medium for a short time** to transmit data. The station replies with data (if it has any), and the process repeats for other stations.

The Access Point (AP) periodically broadcasts **beacon frames**. These beacons contain information about the network, including whether the AP supports PCF. Stations (nodes) in the network use the beacons to discover nearby APs and learn more about the network characteristics.

Before a node can fully connect to an AP, it must first authenticate itself to the AP, and the AP must authenticate itself to the node. Once authentication is successful, the node needs to associate with the AP to formally join the network. The node sends an association request management frame to the AP. The AP responds with an association response. In the association/reassociation frames the node announces to the AP whether it is pollable and capable to transmit during the contention free period (CFP).

SYNCHRONIZATION

TSF (Timing Synchronization Function) is a mechanism used in IEEE 802.11 wireless networks to keep all devices (stations) synchronized to a common clock. TSF is crucial for maintaining proper timing in operations like frame coordination, power-saving modes, and the scheduling of contention-free periods (CFP) in networks that use the Point Coordination Function (PCF).

Infrastructure Beacon Generation refers to the process by which an Access Point (AP) periodically sends beacon frames to announce its presence and provide crucial network information to connected stations (devices) and those looking to join the network. Beacon Interval: the time between two consecutive beacons, typically 100ms by default.

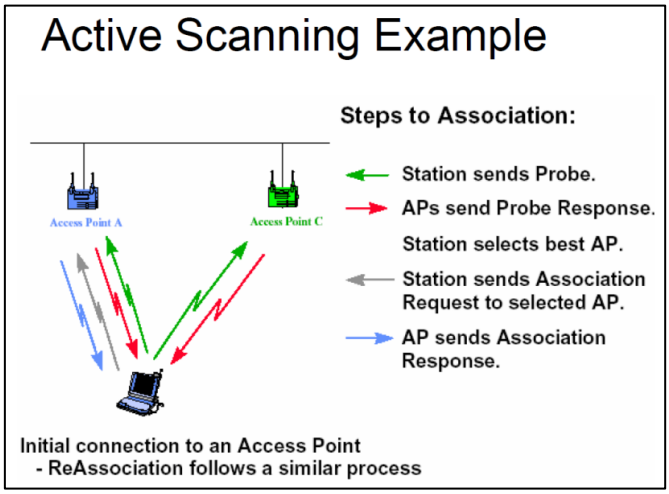
POWER MANAGEMENT APPROACH

Power Management in IEEE 802.11 is a mechanism designed to reduce power consumption for devices. The power management approach **allows devices to enter sleep mode** when not actively transmitting or receiving data, while still maintaining a connection to the network. When a station is in power-saving mode and cannot receive data immediately, the AP will buffer any incoming unicast frames for that station. The **station periodically wakes up** at specified intervals to listen for beacon frames from the AP. The AP includes TIM (Traffic Indication Map) information in the beacons to indicate which stations have buffered data waiting at the AP. If the beacon shows that there is data waiting, the station sends a PS-Poll frame (Power-Save Poll) to the AP to request the buffered data. If the beacon shows no pending data for the station, the station can return to sleep.

SCANNING

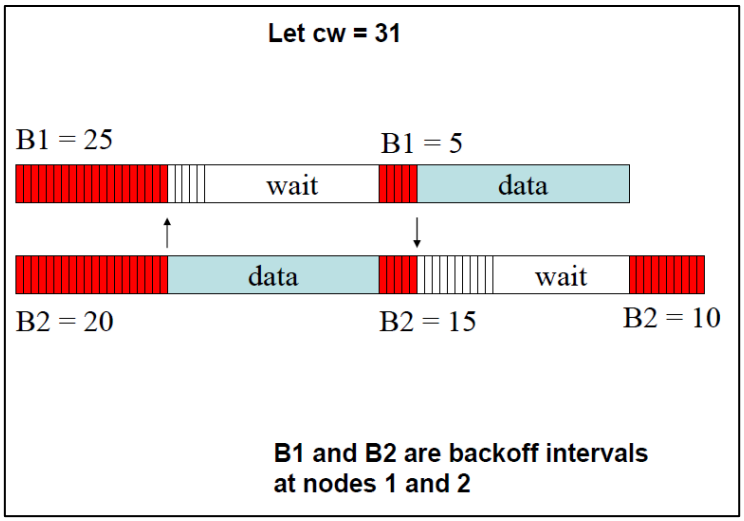
Scanning refers to the process used by stations (client devices) to discover (and join) available wireless networks and their characteristics. There are two main types of scanning: **Passive Scanning** and **Active Scanning**. Both methods are utilized by devices to find access points (APs) and assess the network environment.

In passive scanning, the station listens for beacon frames broadcasted by nearby access points. In active scanning, the station actively sends out probe requests to discover nearby access points. APs that receives these requests respond with probe responses.



CONGESTION AVOIDANCE

The **Distributed Coordination Function (DCF)** is the fundamental access method for handling congestion avoidance in wireless networks. The key mechanism to avoid collisions when multiple stations try to access the same channel is through a random backoff process, which helps ensure that two stations do not transmit simultaneously. Before a station can transmit a data packet (or a RTS frame), it must select a random backoff interval to avoid collisions with other stations that may also want to transmit. The countdown occurs only when the medium is sensed as idle. The station continuously monitors the medium through CSMA to detect whether the medium is busy or idle. If the medium becomes busy (i.e., another station starts transmitting), the backoff timer is paused and then resumed when the medium is idle again.



As we see in the example, B1 has a higher value of backoff than B2, so when B2 starts transmitting B1 backoff pauses, then resumes when the channel is free.
The contention window (CW) is the range from which a station randomly selects its backoff interval. The size of the contention window adjusts dynamically based on the success or failure of transmissions. If success, backoff value is restored to CWmin, otherwise it increases the cw. Before transmitting a packet, a station randomly selects a backoff interval from the range [0, CW]. The CW starts at a small value (e.g., CWmin = 15 slots), and as collisions occur, the CW increases (exponentially) to avoid further collisions. Choosing a large cw leads to large backoff intervals and can result in larger overhead. Choosing a small cw leads to larger number of collisions. The time spent counting down backoff intervals contributes to MAC overhead.

The **MILD algorithm** is used in the MACAW (MACA for Wireless) protocol as an alternative to the more aggressive Binary Exponential Backoff (BEB) algorithm used in IEEE 802.11 (using DCF, when a collision occurs cw is doubled and when a node completes a successful transmission cw is restored to CWmin).
The key idea behind MILD is to provide a more balanced approach to handling collisions and congestion in wireless networks, avoiding abrupt changes in the contention window (CW) size and improving network stability. The core philosophy of MILD is exponential increase and linear decrease:

- Exponential Increase: when a collision occurs, the CW increases faster (multiplying by 1.5), to react to congestion and spread-out access attempts.
- Linear Decrease: after a successful transmission, the CW decreases slowly (subtracting by 1), allowing the system to slowly converge back to lower contention without oscillating too quickly.

FAIRNESS

Fairness is a critical consideration in wireless networks, particularly when multiple nodes or devices are attempting to share the limited bandwidth of a communication medium. The question of fairness arises when determining how to allocate access to the network in a way that ensures equitable treatment of all nodes. The simplest definition of fairness is that **all nodes should receive equal bandwidth**.

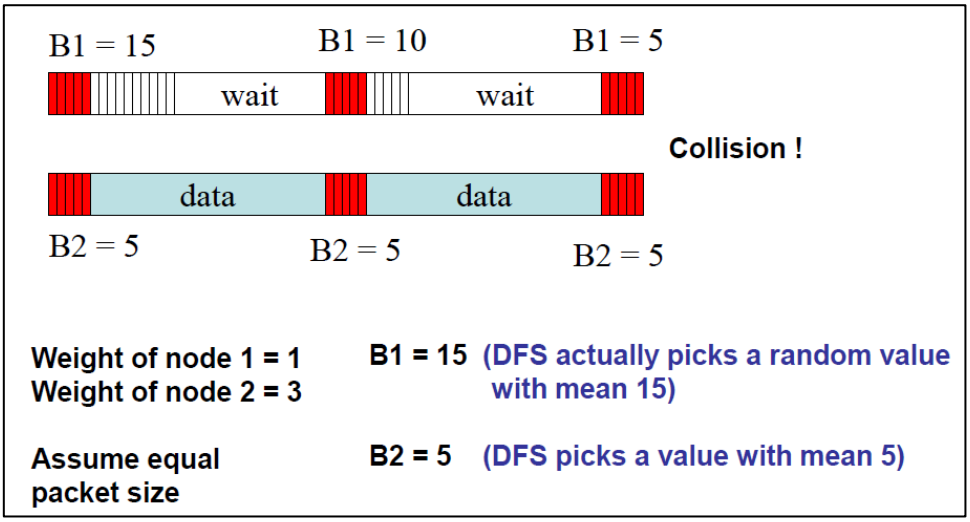
- Assume that initially, A and B both choose a backoff interval in range [0,31] but their RTSs collide
- Nodes A and B then choose from range [0,63]
  - Node A chooses 4 slots and B chooses 60 slots
  - After A transmits a packet, it next chooses from range [0,31]
  - It is possible that A may transmit several packets before B transmits its first packet
- Observation: unfairness occurs when one node has backed off much more than some other node

MACAW fairness solution works by having **nodes share their current contention window values** (when transmitting a packet) with all other nodes that overhear their transmissions. This ensures that all nodes use that CW for their future transmission attempts, equalizing the chances of gaining access to the medium and improving fairness. By resetting all competing nodes to the same CW value, MACAW avoids the unequal access opportunities that can arise from independently managed CW sizes, which is a problem in protocols like IEEE 802.11 DCF.

DISTRIBUTED FAIR SCHEDULING (DFS)

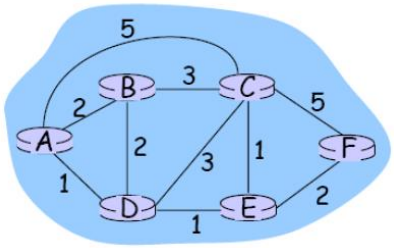
Each node (or flow) is assigned a weight that indicates its priority or required share of the bandwidth. Nodes with higher weights receive a proportionally larger share of bandwidth, allowing for fair yet differentiated access to network resources.

Distributed Fair Scheduling (DFS) is a fully distributed algorithm that achieves weighted fair queuing by adjusting each node's backoff interval based on the ratio of its packet size to its assigned weight. By doing so, DFS ensures that each node receives a fair share of the bandwidth proportional to its weight.



4. Network layer

The main goal of a routing protocol is to **determine the best path for data packets to travel across a network**, from a source to a destination. In a large network, routers use routing protocols to exchange information about the network topology and decide how to forward packets. Routing algorithms can be represented through graph abstractions, where nodes are routers, and the edges are physical links. Typically, a good path is a minimum cost path.



Routing Algorithm Classification

Global or decentralized information?

Global:

- all routers have complete topology, link cost info
- "link state" algorithms

Decentralized:

- router knows physically-connected neighbors, link costs to neighbors
- iterative process of computation, exchange of info with neighbors
- "distance vector" algorithms

Static or dynamic?

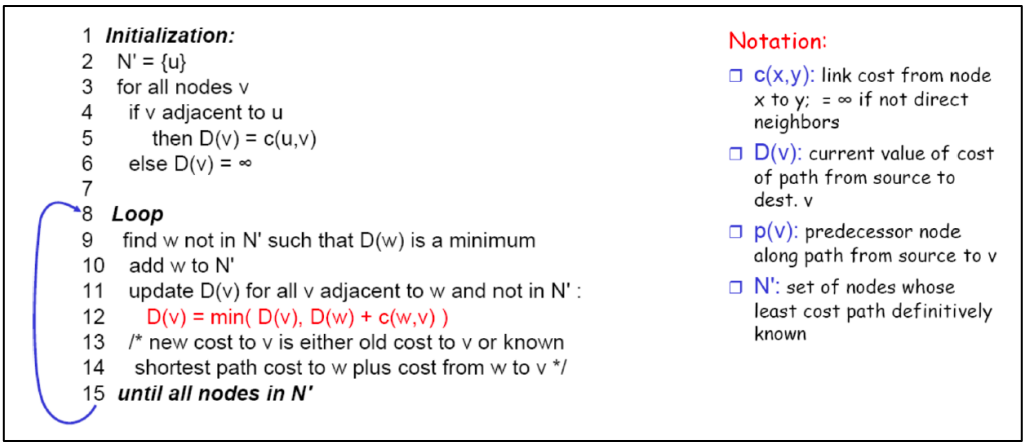
Static:

- routes change slowly over time

Dynamic:

- routes change more quickly
  - periodic update
  - in response to link cost changes

Dijkstra's Algorithm is a widely-used algorithm to find the **shortest path** from a source node to all other nodes in a graph. It operates by progressively exploring the nearest nodes, updating the shortest-known path to each, and finally determining the shortest paths to all nodes. It is the **core of many link-state routing protocols**. In the context of networking, nodes represent routers, and edges (links) represent the connections between them, often weighted by factors like link cost, latency, or bandwidth. The link cost is a value known to all nodes, this is because info from any node is broadcasted to all nodes, so at the end, all nodes "know" the others.



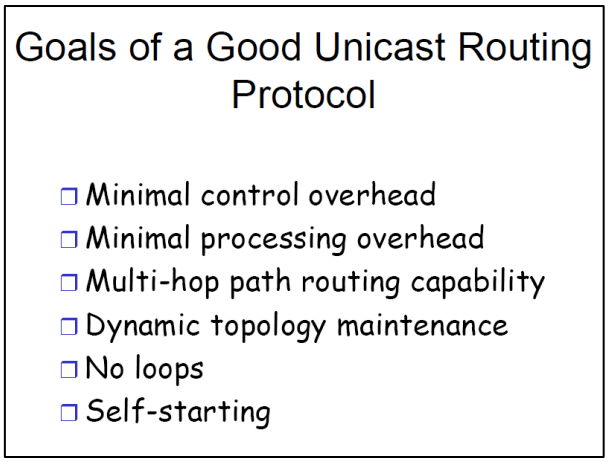
4.1 MANET problems

Traditional routing algorithms, designed for wired networks with fixed infrastructure and infrequent changes, are not well-suited for MANETs. The dynamic and decentralized nature of MANETs introduces unique challenges that make conventional routing protocols inefficient. Traditional algorithms are inefficient due to **slow convergence time**, and non-functional due to large amounts of data and the inability to deal with asymmetric links.
Convergence time refers to the time it takes for all nodes in the network to have a consistent and accurate view of the network's topology after a change.

In traditional networks (like wired networks), the network topology is static or changes very slowly. Routing algorithms based on Link-State or Distance-Vector rely on periodic updates to maintain routing tables (assuming the topology is relatively stable). In MANETs, nodes are mobile, so the **network topology changes frequently and unpredictably**.

MANETs typically operate in wireless environments with **limited bandwidth and higher latency**. Exchange of routing information (control packets) can consume valuable bandwidth, leaving less capacity for actual data transmission. Unlike the data packets that carry actual user information, control packets are essential for maintaining the network's operation, and they are also used to periodically update routing tables. Anyway, **control traffic consumes power** because each packet that a device sends or receives requires transmission or processing. Even though control packets are smaller than data packets, their frequent transmission can significantly drain a device's battery, especially in large or dynamic networks where routing information changes often.

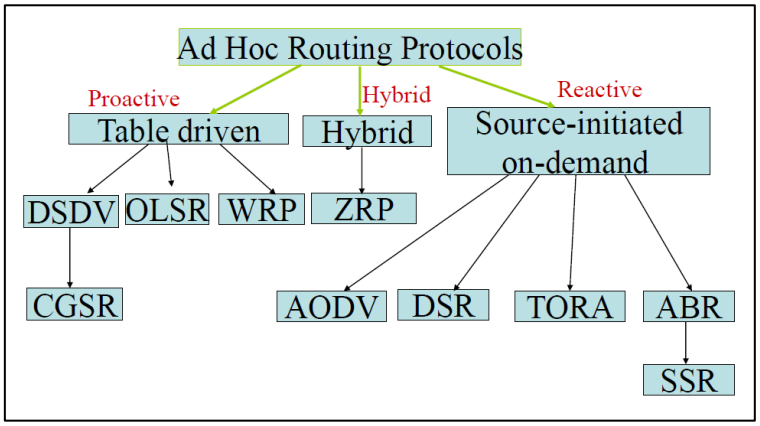
Centralized approaches are too slow and not robust enough for MANET. In MANETs, nodes can move in and out of range frequently, leading to rapid changes in the network topology. Centralized approaches rely on a fixed central controller or server to manage the network, which can struggle to keep up with these changes. In a MANET all or most all nodes can work as routers. The only **"path length" is a straightforward metric for MANET** since it fails to capture the complexities and challenges of routing in MANETs. A more effective routing strategy should consider a combination of metrics, such as link quality, latency, bandwidth, energy efficiency, and network load.



4.2 Routing protocols

Routing protocols can be classified into different categories based on how they manage routing information and establish paths in a network. The three main classifications are:

1. **Table-driven (proactive)**: this approach is based on traditional distance-vector and link-state protocols. Every node in the network has a routing table that contains the best routes to all other nodes. The routing overhead is independent of the actual route usage. This means that routing information is exchanged periodically, regardless of whether the routes are currently in use or not. Since routes are always available, there is no delay for route discovery when sending packets. Changes to the network topology are immediately propagated. *Note*: mobility results in significant update (not always necessary if the network moves faster than the route request).
2. **Source-initiated (reactive)**: these protocols establish routes only when needed by flooding (On-Demand Route Discovery). Routes are maintained only for active communications. If the route is no longer needed, it can be discarded. The source node initiates a route discovery process, which involves broadcasting a request to discover a path to the destination. Typically, this approach has less control overhead and better scaling properties. As a drawback it can have a long delay in finding the route. Route maintenance is used to repair routes.
3. **Hybrid protocols**: utilize both proactive and reactive strategies to optimize routing efficiency.



4.2.1 Proactive approach

DESTINATION SEQUENCED DISTANCE VECTOR (DSDV)

DSDV builds on the principles of the **Bellman-Ford algorithm**, which calculates the shortest paths from a source node to all other nodes in the network. Each node maintains a routing table that contains the best-known distance (or cost) to each destination node.
To maintain consistency and avoid routing loops, DSDV uses **sequence numbers**. Each route entry in the routing table is associated with a sequence number generated by the destination node. Since sequence numbers increments over time, routes with higher sequence numbers are considered fresher or more up-to-date. Nodes will prefer routes with higher sequence numbers when making routing decisions (when two equal routes are received from a neighbour, the one with the smallest hop count is selected).
DSDV optimizes routing updates by employing **incremental data exchange**. Instead of broadcasting the entire routing table, nodes only send updates for the routes that have changed. This minimizes the amount of control traffic generated and conserves bandwidth.

Routing table updates create lots of control traffic, DSDV addresses this problem by using types of routing update packets. This strategy helps to minimize unnecessary bandwidth consumption while maintaining up-to-date routing information across the network:

1. **Full Dumps**: These packets contain the complete routing table of the sending node and due to that they are transmitted relatively unfrequently.
2. **Incremental Updates**: only the changes in the routing information since the last update are transmitted. If the updates are “too large” (i.e. do not fit into a NPDU - Network Protocol Data Unit) the full dump is sent.

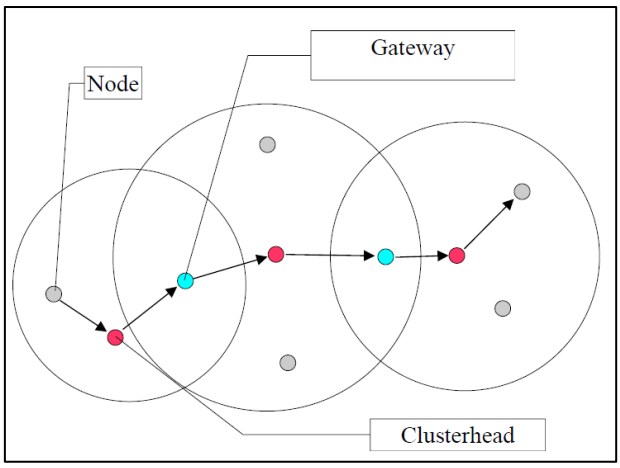
OPTIMIZED LINK STATE ROUTING PROTOCOL (OLSR)

This protocol is good for large and dense networks and is great for the traffic patterns where there are multiple mobile hosts communicating each other.
OLSR is based on a **link state routing mechanism**, where each node maintains a complete view of the network topology. This is achieved through the dissemination of link state information among nodes.
One of the core optimizations in OLSR is the use of MultiPoint Relays (MPRs). Each node selects a subset (that can change over time) of its neighbours as **MPRs to reduce the number of retransmissions needed for routing updates**. When a node needs to broadcast control information (like topology changes), it sends the message only to its selected MPRs. This minimizes the flooding of broadcast packets in the network. The subset of neighbours is indicated by the selector nodes in their hello messages*. A node chooses its MPRs such that every two-hop neighbour (nodes that are not direct neighbours but can be reached through a single hop) is covered by at least one of its MPRs. This ensures that control messages can be disseminated effectively without overwhelming the network with broadcasts.

*HELLO messages: sent periodically by each node to discover its neighbours and determine the status of links to those neighbours.

CLUSTER GATEWAY SWITCH ROUTING (CGSR)

To facilitate communication and reduce overhead CGSR organizes the network into **clusters**, each managed by a designated cluster head**,** selected using an election, usually the node that has the highest connectivity (i.e., the most neighbours) or other selection criteria (like battery power or node ID) is often chosen as the Cluster Head (CH).



The CH serves as a coordinator for a cluster, it is responsible for managing communications within its cluster. Different clusters are connected through a gateway node. **Nodes send packets through the CH which uses DSDV to communicate with other CH**.
Nodes periodically share their routing information with their Cluster Heads and gateway nodes, allowing for dynamic updates of the routing tables.

4.2.2 Reactive approach

AD HOC ON-DEMAND DISTANCE VECTOR ROUTING (AODV)

AODV employs **destination sequence numbers** to ensure loop freedom in route discovery and maintenance. Each route entry in the routing table is associated with a sequence number, which is updated whenever the destination node changes. Higher sequence numbers indicate more recent routes, helping to prevent routing loops. Two routing tables are utilized to store information, one is for unicast routes and one for multicast. Information that are stored in routing table are: *dest_address, next-hop_address, dest_seq_num, life_time* (how long the route is valid, if not used it expires). For each destination, a node maintains a list of its precursor nodes, to route through them.
Unlike traditional distance vector protocols, AODV limits routing table exchanges to occur only along a given route. This means that instead of constantly broadcasting the entire routing table, AODV only shares routing information between nodes that are involved in an active communication process.

When a source node needs to send data to a destination for which it does not have a valid route in its routing table, it broadcasts via a technique called flooding a **Route Request Packet (RREQ)** to all the neighbour to discover a route. This packet contains information about node's

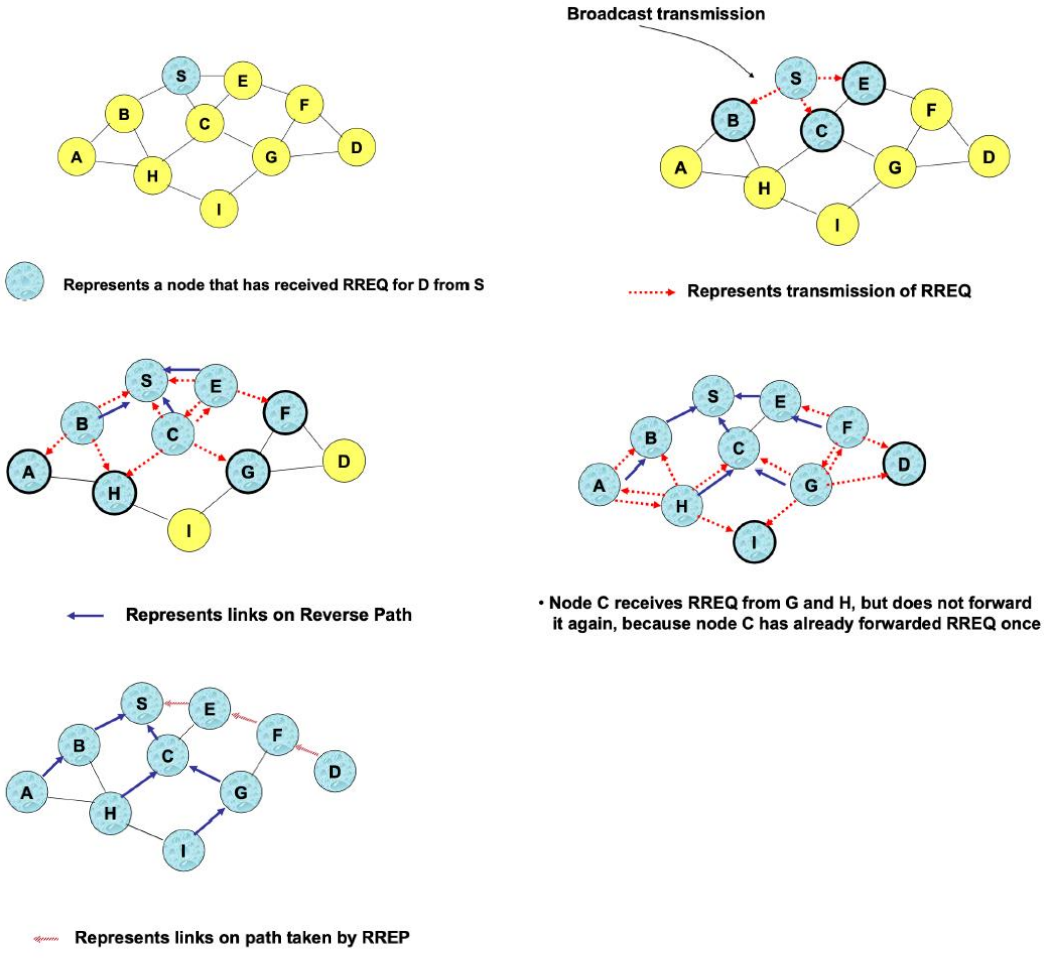
IP address, source node sequence number, destination IP address, destination sequence number, ***broadcast ID number*** (that gets incremented each time a source node uses RREQ). If a node discovers that a route is no longer valid (e.g., due to a node moving out of range), it can generate a **Route Error Packet (RERR)** to inform the affected nodes.

Once an intermediate node receives a RREQ it sets up a **revers route entry** for the source node in its route table. The reverse route allows the intermediate node to know how to send a **Route Replay Packet (RREP)** or any other packets back to the source of the RREQ.

When the RREQ reaches the destination node, to respond with a RREP it should have in its routing table an unexpired entry for the destination, and the sequence number of the destination at least as great as in RREQ. If these two requirements are satisfied, and the IP address of the destination is the same of the node we are looking for, the destination node sends back to the source a RREP using unicasting and not flooding.

When an intermediate node receives a RREQ, it can reply with a RREP instead of forwarding the RREQ to other nodes only whether it has a valid and sufficiently up-to-date route to the destination. To determine that just compare the destination sequence number in the RREQ with the one stored in the routing table, if higher sends back the RREP. Anyway, the probability that an intermediate node forwards a RREP is not so high as in DSR (as we will see later).

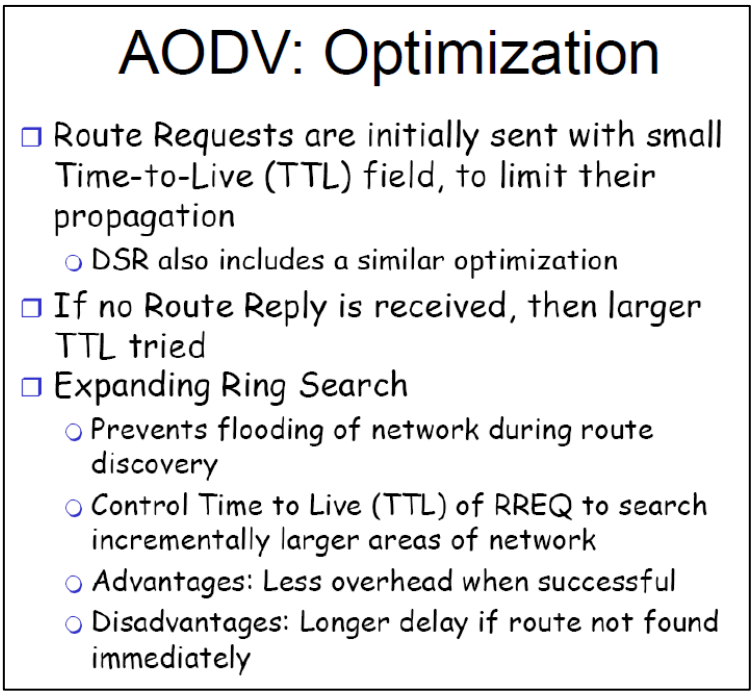
This is an example of route discovery process:



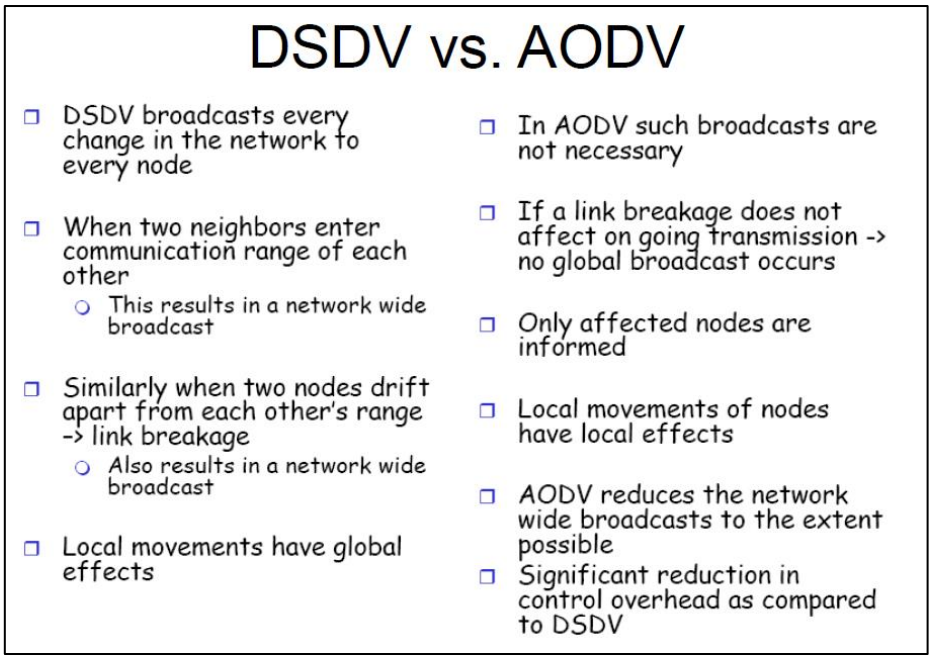
AODV timeouts represent for how much time a route is held in the routing table before getting discarded. The reverse path is set up when a node receives a RREQ. The *reverse path* helps guide the RREP back to the source once the route to the destination is found. However, this reverse path is temporary, and if no RREP is received within a certain time frame, the entry is discarded. Once a *forward path* is established (either by receiving a RREP or during data transmission), the entry is maintained in the routing table for active use. However, forward path entries are not kept indefinitely and are removed after a certain period of inactivity (for the duration of the ***active_route_timeout interval***).

Link failure detection

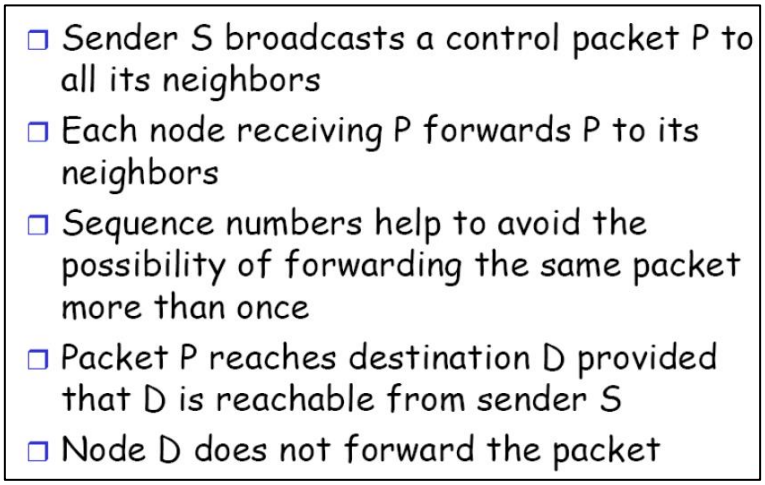
Due to the mobility of nodes, links between them can often break, requiring the network to detect these failures promptly to avoid sending data through broken paths. Nodes periodically send small control packets (**HELLO messages**) to their neighbours to indicate their presence. If a node stops receiving HELLO messages from a neighbour after a certain time, it assumes the link to that neighbour has failed.



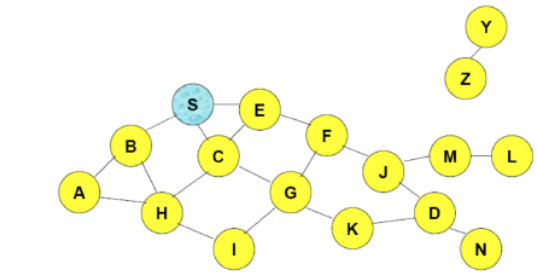
In a larger network with many nodes, if we need to find a path between two nearby nodes, flooding can be inefficient. It takes time and may interfere with other communications, as multiple nodes trying to communicate simultaneously can cause disturbances.
A more efficient approach is **incremental flooding**: starting with a limited flooding range and gradually increasing it if the destination is not reached. Although this step-by-step approach may still take time, it reduces interference and is generally more effective in such scenarios.



Flooding for Control Packet Delivery is a technique used in networking, particularly in ad hoc networks and mobile ad hoc networks (MANETs), to disseminate control information efficiently throughout the network.

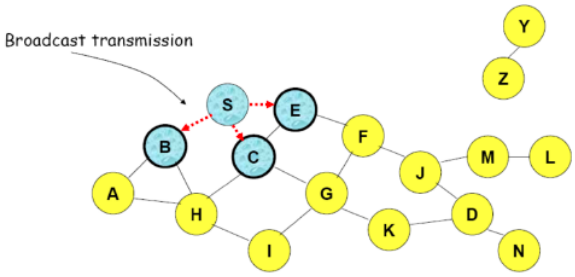


Flooding for Control Packet Delivery example:



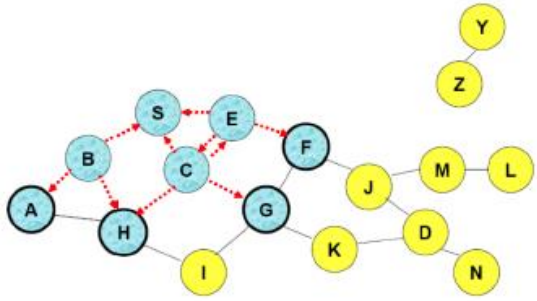
Represents a node that has received packet P

Represents that connected nodes are within each other's transmission range

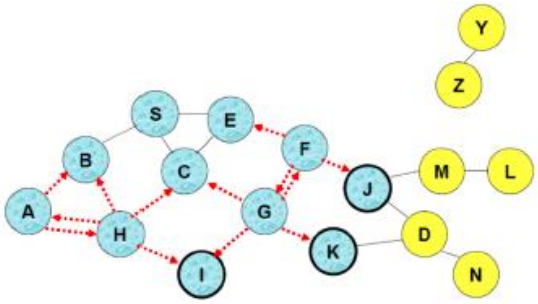


Represents a node that receives packet P for the first time

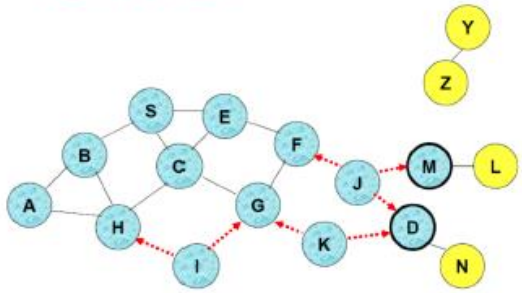
Represents transmission of packet P



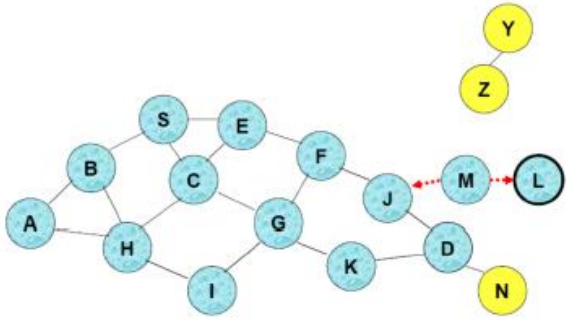
❑ Node H receives packet P from two neighbors:
potential for collision



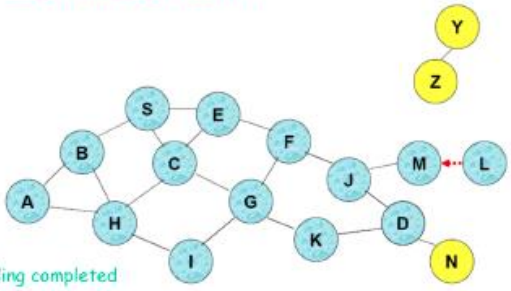
❑ Node C receives packet P from G and H, but does not forward it again, because node C has already forwarded packet P once



❑ Nodes J and K both broadcast packet P to node D
❑ Since nodes J and K are hidden from each other, their transmissions may collide
=> Packet P may not be delivered to node D at all, despite the use of flooding



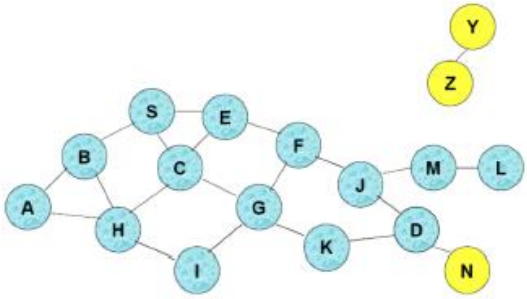
❑ Node D does not forward packet P, because node D is the intended destination of packet P



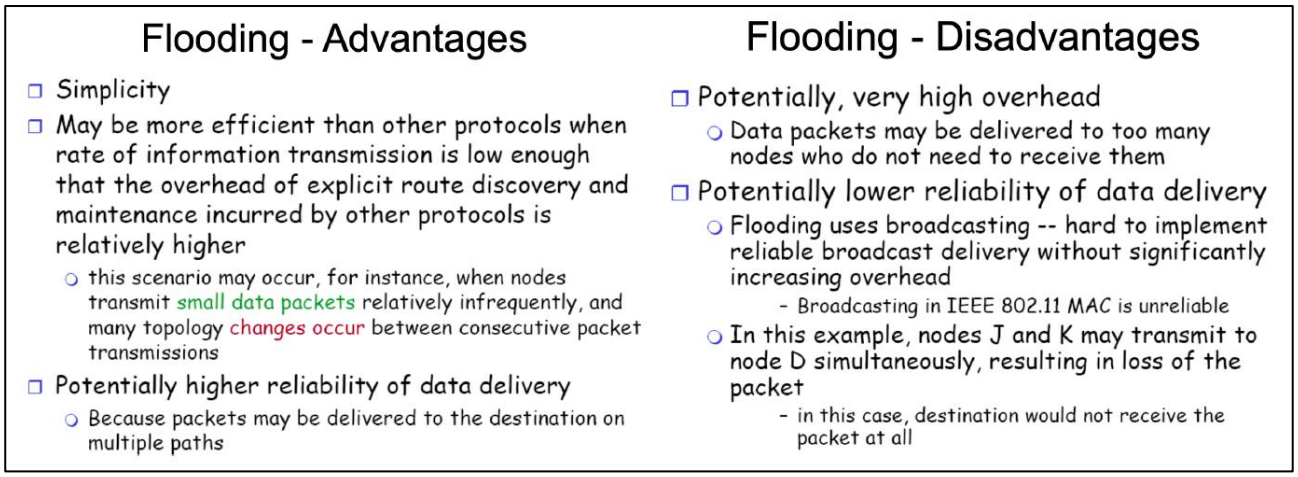
❑ Flooding completed

❑ Nodes unreachable from S do not receive packet P (e.g., node Z)

❑ Nodes for which paths go through the destination D also do not receive packet P (example: node N)



❑ Flooding may deliver packets to too many nodes (in the worst case, all nodes reachable from sender may receive the packet)



While flooding has several advantages, such as robustness, simplicity, and efficiency in disseminating information, it also has drawbacks, including potential congestion and redundant packet transmission. Therefore, it is often used selectively in conjunction with other routing strategies to balance the benefits and mitigate the downsides.

DYNAMIC SOURCE ROUTING (DSR)

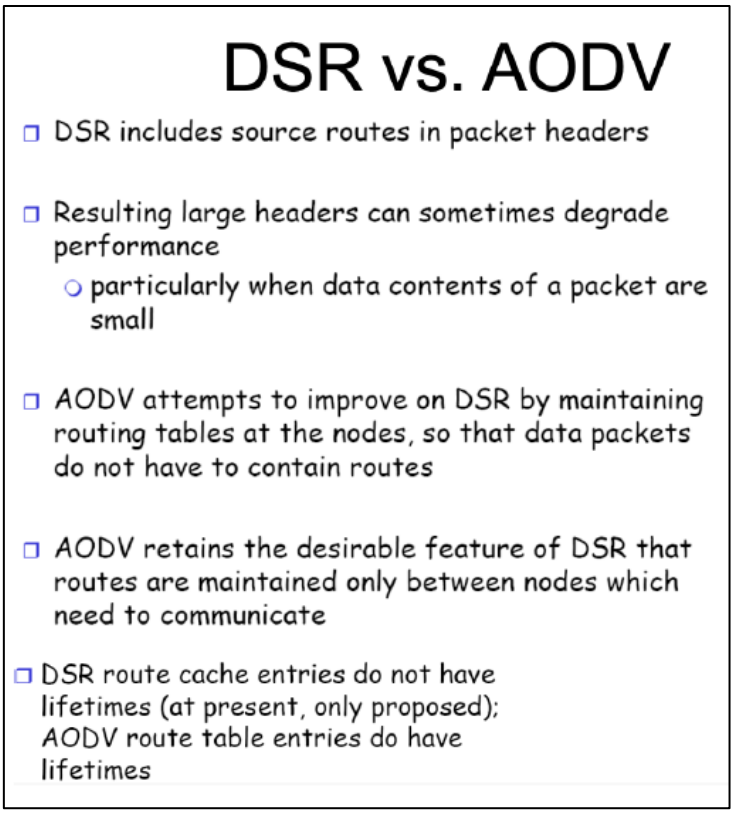
DSR is an on-demand routing protocol, meaning that routes are established only when needed, rather than always maintaining a complete routing table.

The protocol uses source routing, where **the complete route to the destination is included in the packet header**. Each node that receives the packet can see the entire path, allowing it to forward the packet to the next hop.

DSR maintains **routing table entries only at the source node**, reducing overall routing overhead. Intermediate nodes do not keep full route tables for every destination.

Each node in the DSR protocol maintains a **route cache**. This cache stores recently discovered routes, allowing nodes to quickly access known paths without needing to initiate a new route discovery process.

When a source node needs to communicate with a destination node for which it has no available route in its cache, it generates a RREQ. The source node broadcasts the RREQ packet to its neighbours. As the RREQ travels through the network, each intermediate node that receives the RREQ appends its own address to the packet before forwarding it. Once the RREQ reaches the destination node, the packet contains the complete route from the source to the destination. The destination node then processes the RREQ and prepares a RREP packet.



ASSOCIATIVITY BASED ROUTING (ABR)

Instead of using traditional metrics like the shortest hop count to determine the best route between nodes, ABR relies on a new metric called the **Degree of Association Stability**. This metric is used to measure the stability of links between nodes based on how long they have been associated with each other. The longer two nodes have stayed connected, the more stable their link is considered. ABR assumes that **less mobility/better links** provide more stable and reliable routes, as they are less likely to move away and cause a break in the communication link.

SIGNAL STABILITY ROUTING (SSR)

A routing protocol designed for MANETs, where **signal strength** is used as the primary metric for selecting routes. Similar to Dynamic Source Routing (DSR), SSR is an on-demand routing protocol, but it places special emphasis on the quality of the signal strength when deciding which paths to use. Nodes monitor the signal quality of incoming packets to decide if a link is "strong" or “weak."

SSR uses a route discovery process similar to DSR, where a source node broadcasts a Route Request (RREQ) to find a route to the destination. As in DSR, intermediate nodes add their addresses to the RREQ packet before forwarding it. What distinguishes SSR is that intermediate nodes only forward the RREQ packet if the signal strength over the link through which they received it is strong.

OTHER METRICS

In addition to signal strength, routing protocols in mobile ad-hoc networks (MANETs) can use various other metrics to evaluate the quality of paths. Two important metrics that have emerged in this context are **Expected Transmission Time (ETT)** and **Weighted Cumulative Expected Transmission Time (WCETT)**. These metrics provide more nuanced evaluations, particularly in multi-hop, multi-radio, and asymmetric networks.

The ETT metric measures the expected time needed to successfully transmit a packet over a link, considering the quality and capacity of that link.

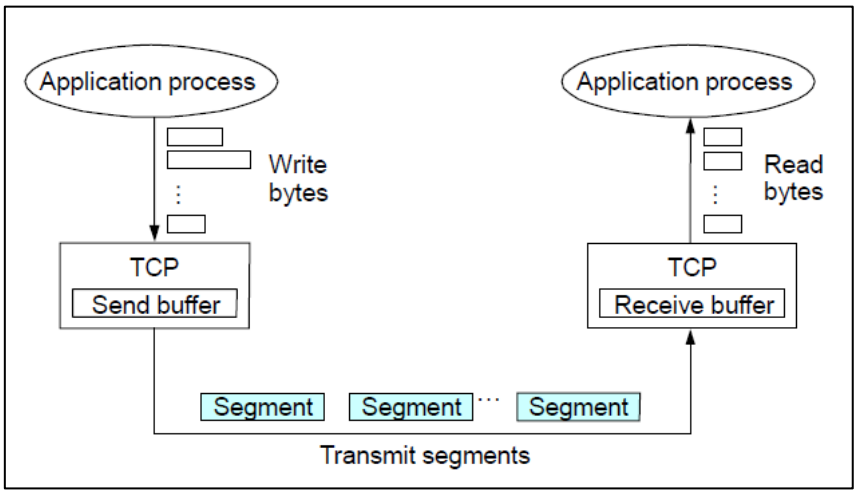
WCETT is an extension of ETT and is particularly suited for networks with multiple radios or asymmetric rate links. WCETT helps in balancing the load across multiple channels in a multi-hop network.

At the end consider that:

- Proactive routing protocols suitable for high traffic load, low mobility.
- On-demand routing protocols suitable for low traffic load and/or moderate mobility.
- With high mobility, flooding of data packets may be the only option.

5. Transport layer

TCP (Transmission Control Protocol) is a specific protocol used for data transmission over a network. It operates primarily at the Transport Layer of the OSI model. It is the most widely used Internet protocol, (web, peer-to-peer, FTP, telnet...). In particular, TCP operates on top of the Internet Protocol (IP), which handles packet routing and addressing. While IP ensures packets are sent across the network, TCP ensures the **reliability** and **order** of these packets once they reach their destination.



TCP features:

- **Connection-oriented**: TCP establishes a connection between the sender and receiver before data can be transmitted. This connection is set up using a three-way handshake process (SYN, SYN-ACK, ACK) to ensure that both parties are ready for data transfer.
- **Full-duplex communication**, allowing data to be sent and received simultaneously between two endpoints.
- **Reliable data transfer**: it guarantees that data sent from one end is received correctly by the other end, ensuring that lost packets are retransmitted, and data is delivered in correct order.
- **Byte-stream orientation**: TCP treats data as a continuous stream of bytes, rather than discrete packets. Applications write data in bytes to the TCP layer, and TCP transforms this byte stream into manageable chunks (segments) for transmission.
- **Flow control**: flow control mechanisms in TCP prevent the sender from overwhelming the receiver with too much data too quickly. TCP uses a ***sliding window protocol*** where the sender can only send a certain amount of data (the size of the window) before needing an acknowledgment from the receiver. If the receiver's buffer is nearing its capacity, it can inform the sender to slow down or stop sending data until it has processed some of the received data.
- **Congestion control**: helps manage the overall data traffic in the network to prevent packet loss and ensure efficient transmission by preventing the transmitter to overwhelm the network.

Traditional TCP protocols, although highly effective in wired network environments, encounter several performance challenges in wireless scenarios. This is largely due to the unique characteristics of wireless networks, which include higher packet loss rates primarily caused by interference and signal degradation, leading to reduced data integrity and retransmissions. Moreover, wireless networks experience variable latency as network conditions fluctuate, affecting the consistency of data transfer rates. Furthermore, these protocols are not well-optimized for managing bursty data flows, which are common in wireless networks.

RELIABILITY

In TCP, reliability is a critical feature designed to ensure that data is delivered accurately.

- **Checksum for error detection**: each TCP segment includes a checksum in its header, the checksum is used to detect any bit-level errors that may occur during transmission. If the receiver detects that the checksum does not match the data, it discards the segment and expects the sender to retransmit it.
- **Sequence number used for Sequencing and Duplicate Detection**: sequence numbers are assigned to each byte in the data stream. Each TCP segment includes the sequence number of the first byte in that segment. These sequence numbers help the receiver put the incoming segments back in the correct order. If packets are lost, reordered, or duplicated during transmission, TCP can detect and handle those errors by retransmitting the lost packets and reordering the out-of-sequence packets.
- **Packet loss and Retransmission**: TCP uses timeouts to detect packet loss. If the sender does not receive an acknowledgment (ACK) for a packet within a certain period (known as the RTO, or Retransmission Timeout), it assumes the packet was lost and retransmits it.

  TCP can also detect packet loss through a faster mechanism known as duplicate acknowledgment. If the receiver notices a gap in the sequence of received segments (e.g., packet loss), it sends duplicate ACKs for the last correctly received packet. When the sender receives three or more duplicate ACKs, it interprets this as an indication that a packet has been lost and immediately retransmits the lost segment without waiting for the RTO. This process is part of TCP's fast retransmit mechanism.

TIMEOUT-BASED RECOVERY

A crucial part of this mechanism is determining the appropriate **Retransmission Timeout (RTO)**. If a packet is not acknowledged within the calculated RTO, TCP assumes it has been lost and retransmits it. The challenge lies in accurately determining this timeout period.

Round Trip Time (RTT) is the time it takes for a packet to travel from the sender to the receiver and for the acknowledgment (ACK) to come back. TCP needs to wait for at least one RTT before deciding whether to retransmit a packet. This ensures the sender does not unnecessarily retransmit packets that may just be delayed in the network. **RTT estimation is crucial for determining the right RTO**. If the RTO is set too low, TCP will retransmit too quickly, causing unneeded retransmissions. If the RTO is set too high, TCP will wait too long to retransmit, leading to poor throughput as lost packets are not detected and retransmitted quickly enough.

The RTO should change as the RTT changes (e.g., when the network becomes more congested or less congested). However, the adaptation should not be too fast (which could cause instability), nor too slow (which could cause poor performance). A commonly used heuristic is to set the RTO to be about 4 times the current estimated RTT. If the RTO expires (meaning that a packet is lost) and a new timeout is set for the packet retransmission, this timeout is doubled and so it becomes 8RTT, 16RTT and so on.

FAST RETRANSMIT & FAST RECOVERY

The key idea is to take advantage of **duplicate acknowledgments (dupacks)** as a signal for potential packet loss. A dupack is an acknowledgment (ACK) that is sent for the same data segment multiple times. Normally, TCP sends an ACK for each segment it receives. If a segment is lost, the receiver continues to acknowledge the last segment it received in sequence. This results in multiple duplicate ACKs for the same segment.
For example, if the sender sends segments 1, 2, 3, 4, and segment 2 is lost, the receiver would ACK segment 1 again even after receiving packets 3 and 4 (since it is still waiting for packet 2). Duplicate ACK's can occur when a packet is lost, when packets need reordering or if the flow control window has changed.
TCP assumes that packet re-ordering is infrequent and not significant enough to cause more than a couple of dupacks. Hence, **TCP waits until it receives three or more duplicate ACKs** for the same data segment **before assuming packet loss**.

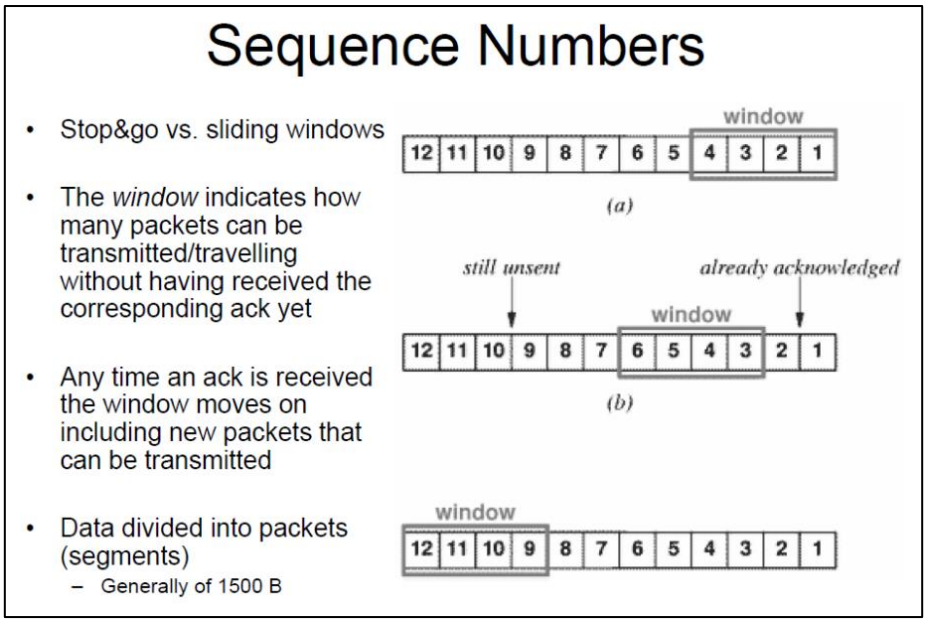
Problem: **coarse-grained TCP timeouts lead to idle periods**. When packet loss occurs and TCP relies on a timeout to retransmit the lost packet, the sender remains idle during this waiting period. This can lead to poor performance, especially when the timeout is too high.

Solution: **fast retransmit & fast recovery** are mechanisms designed to improve performance by reducing idle time and allowing quicker recovery from packet loss.

Fast Retransmit has been explained previously, it uses three duplicate ACKs (dupacks) to trigger retransmission.

Fast Recovery is used to prevent TCP from drastically reducing its sending rate after fast retransmit. After a packet loss is detected and three dupacks are received, TCP performs a fast retransmit of the missing packet, bypassing the normal timeout procedure. Then the sender sets the congestion window (cwnd) to **ssthresh** (**slow start threshold**) instead of dropping the congestion window to one segment per time (initially), as happens after a timeout. This allows the sender to continue transmitting at a reduced, but reasonable, rate rather than drastically slowing down. The sender sends ssthresh new packets to keep the network busy and continue the transmission. TCP continues with additive increase (increasing the window linearly) until the network recovers and normal transmission resumes.

Note: the congestion window represents the maximum number of packets (segments) that is possible to transmit without having received the corresponding ACK yet. The congestion window over the RTT determines the transmission bandwidth.



In **stop&go protocols** (like Stop-and-Wait), the sender transmits one packet and waits for an acknowledgment (ACK) before sending the next packet.
Sliding window, instead, is a more efficient method, allowing multiple packets to be "in flight" (i.e., transmitted but not yet acknowledged). TCP assigns a sequence number to every bite of data in the data stream. If the sender sends a segment starting with byte 1000, and the segment contains 1000 bytes of data, the next segment will start with sequence number 2000, and so on. The receiver sends an ACK with the sequence number of the next expected byte. For example, if the receiver successfully receives bytes 1000 to 1999, it will send an ACK for sequence number 2000, indicating that it is expecting the next byte to start at 2000.

TCP FLOW CONTROL

Flow Control is a mechanism designed to prevent the sender from overwhelming the receiver with too much data. The receiver maintains a receive buffer, which temporarily holds incoming data before the application on the receiver side processes it.
The **Advertised Window** is a field in the TCP header of the ACK packets sent back to the sender which indicates the amount of buffer space available for receiving more data.
The sender has a **Sending Window**, which limits how much data it can send before needing an acknowledgment from the receiver. The Sending Window is the minimum of two values: sender's computed congestion window and advertised window.

Delay-Bandwidth Product is a measure of the capacity of network path between sender and receiver. It represents the maximum number of data that can be traveling on the link at any time, similar to the amount of water filling up a pipe.

- **Delay (or Latency/RTT)**: the time taken for data to travel from sender to receiver.
- **Bandwidth**: the data-carrying capacity of the link (usually measured in bits per second).

⇨ **Delay x Bandwidth** represents the total amount of data that can occupy the link at any given time.

TCP CONGESTION CONTROL

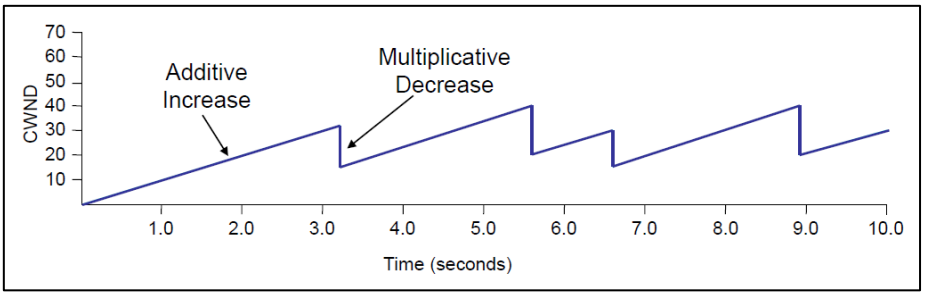
Congestion control is designed to prevent the sender from overloading the network (the sender transmits message at maximum speed until it detects a congestion). TCP operates under the assumption that network devices are “best-effort” routers, meaning they forward packets in a FIFO order without guaranteeing any quality of service. Each TCP source dynamically determines the network’s capacity by observing how quickly acknowledgments return. The ACKs serve as implicit feedback on the network's congestion state. The ACKs "pace" the data transmission, essentially acting as a self-clocking mechanism. When the network is less congested, ACKs return faster, allowing the sender to increase its transmission rate.

The **Additive Increase Multiplicative Decrease (AIMD)** mechanism in TCP congestion control allows TCP to adapt its sending rate based on network conditions. AIMD aims to adjust TCP’s CongestionWindow (CWND) size to reflect the changing available capacity in the network (recall SendingWindow = min{CongestionWindow, AdvertisedWindow}). AIMD idea is to increase CWND when congestion goes down and decrease it when congestion goes up.

- Question: how does the source determine whether or not the network is congested?
- Answer: timeout/dupacks
  - timeout signals that a packet (or more than one) was lost
    - probably with some serious congestion or other problem (disconnection?)
  - Three dupacks signals that a packet was lost but others are still passing through
    - Probably a minor congestion or packets that are seldom lost due to transmission error
  - In general it is assumed that lost packet implies congestion
    - Not true in wireless environments!

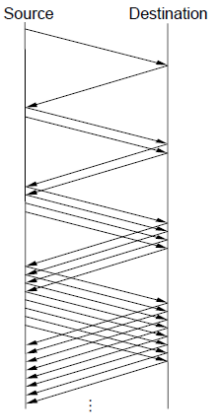
AIMD algorithm

- **Additive Increase (Linear Growth)**: increases the CongestionWindow by one packet per RTT. This means that each RTT without congestion will result in a slightly larger cwnd, gradually allowing more packets in flight. Practically, cwnd+1 when an ACK is returned.
- **Multiplicative Decrease (Fast Reduction)**: if congestion is detected (e.g., by a timeout or receiving three duplicate ACKs), the CongestionWindow is cut in half to quickly reduce the sending rate, thereby relieving network congestion.



Slow Start

- Objective: quickly determine the available capacity in the first part of a connection
- Idea:
  - begin with `CongestionWindow` = 1 pckt
  - double `CongestionWindow` each RTT (increment by 1 packet for each ACK)
  - This is exponential increase to probe for available bandwidth
  - Up to half of cwnd may get lost (when congestion level is reached)
- Used…
  - when first starting connection
  - when connection goes dead waiting for timeout
- SSTHRESH (slow start threshold) indicates when to begin additive increase phase



SSTHRESH and CWND

- SSTHRESH typically very large on connection setup
- Set to one half of `CongestionWindow` (CWND) on packet loss
  - So, SSTHRESH goes through multiplicative decrease for each packet loss
  - If loss is indicated by timeout, set `CongestionWindow = 1`
  - If loss is indicated by 3 dupacks, set `CongestionWindow` equal to half of the congestion window value prior to the loss event
- After loss, when new data is ACKed, increase CWND
  - Manner depends on whether we are in slow start (exponential) or congestion avoidance (linear)

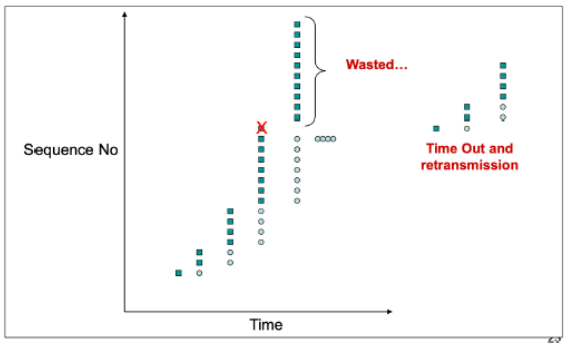
SSTHRESH is a key parameter used in congestion control to determine whether the connection should be in slow start mode or congestion avoidance mode. At the beginning cwnd is equal to 1 (slow start mode) and once it reaches ssthresh value TCP switches to congestion avoidance mode using AIMD algorithm to control congestion on the network.

Legacy TCP versions: TCP Tahoe, TCP Reno, TCP New Reno, TCP SACK, TCP Vegas.

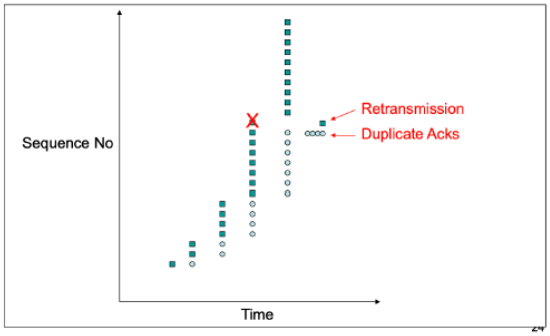
TCP Tahoe Overview

- Standard TCP functions
  - connections, reliability, etc.
- Slow Start
- Congestion control/management
  - Additive Increase/ Multiplicative Decrease (AIMD)
  - Only timeouts to detect losses:
    meaning that after any loss it sets the cwnd to 1, starting so from the beginning

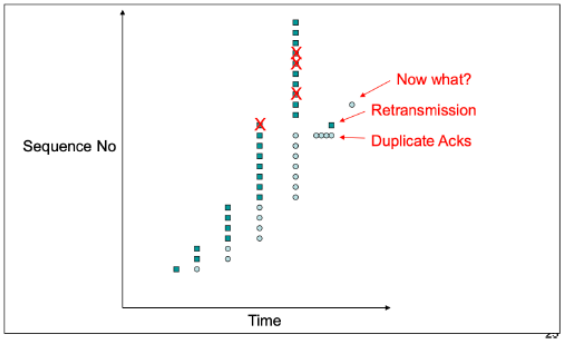
TCP Tahoe



Fast Retransmit



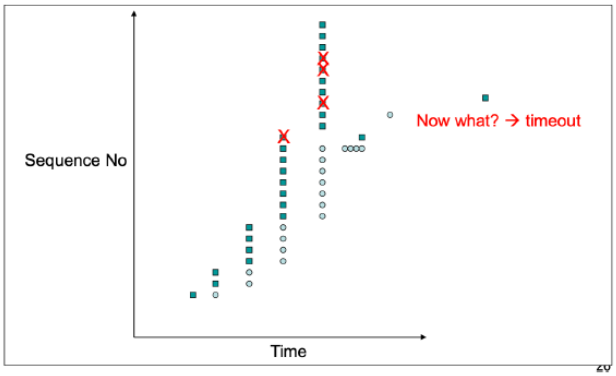
Multiple Losses



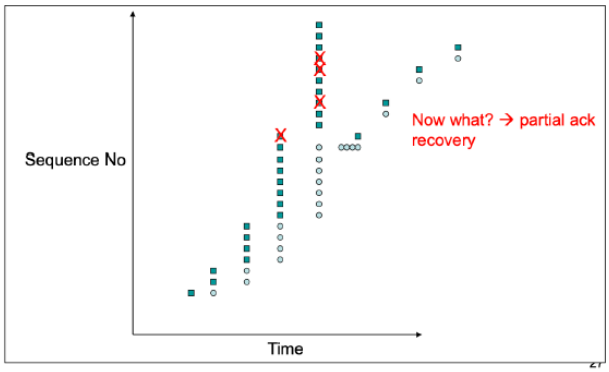
TCP Reno & New Reno

- Fast Retransmit/Fast Recovery
  - Three dupacks to quickly recover from light congestion (1 pkt loss)
- TCP Reno can recover from 1 pkt loss without having a time out
- TCP New Reno
  - Introduces partial ACKs to recover more pkts without resorting to timeouts

TCP Reno



TCP New Reno



<u>TCP RENO</u>: if multiple packets are lost within the same cwnd, it only retransmits the first lost packet due to the 3 dupacks. However, if there's another loss Reno doesn't receive enough dupacks for the next loss, relying so on a timeout to handle it.

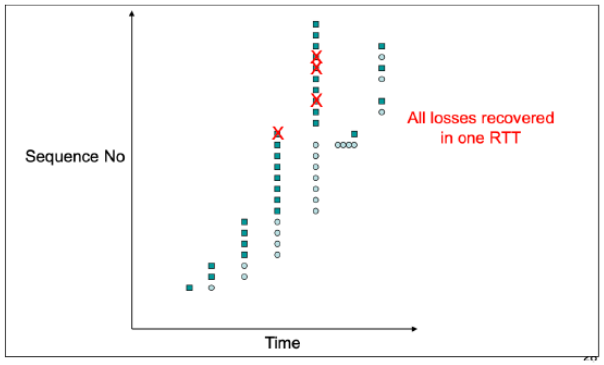
<u>TCP NEW RENO</u>: it introduces the partial ACKs which allow to advertise the received packet until a certain point. The next packet which is not advertised in the partial ACK is retransmitted. Partial ACKs work with the same logic of 3 dupacks, allowing to retransmit the packets immediately without relying for the timeout to handle losses.

TCP SACK

- TCP New Reno can retransmit only 1 pckt every RTT
  - Needs a partial ack to come back
- TCP SACK (Selective Acknowledgment)
  - Returning acks declares which packets (even non contiguous) were received
  - All non received packets can be retransmitted
    - Recover from multiple losses in just one RTT

Particularly suitable for wireless environments, due to the multiple retransmissions in a single RTT

TCP SACK



Instead of informing the sender saying, “all packet received till this one” (partial ACK – New Reno), the idea of SACK is to inform the sender about all the packets that were received (even non-contiguous). In this way, SACK allows the sender to retransmit the missing packets (the ones not declared in the selective ACK) all together in a single RTT. New Reno, instead, sends only the next missing packet in the partial ACK, resulting in 1 retransmission per RTT.

Loss Recovery: Summary

- Two ways to detect losses
  - Time outs
  - Three dupacks
- With timeout expiration
  - ssthresh = cwnd / 2
  - cwnd = 1 (so, restart in slow start phase)
- With three dupacks
  - ssthresh = cwnd / 2
  - cwnd = cwnd / 2 (so, restart in cong. avoidance phase)

Congestion Control Functionality

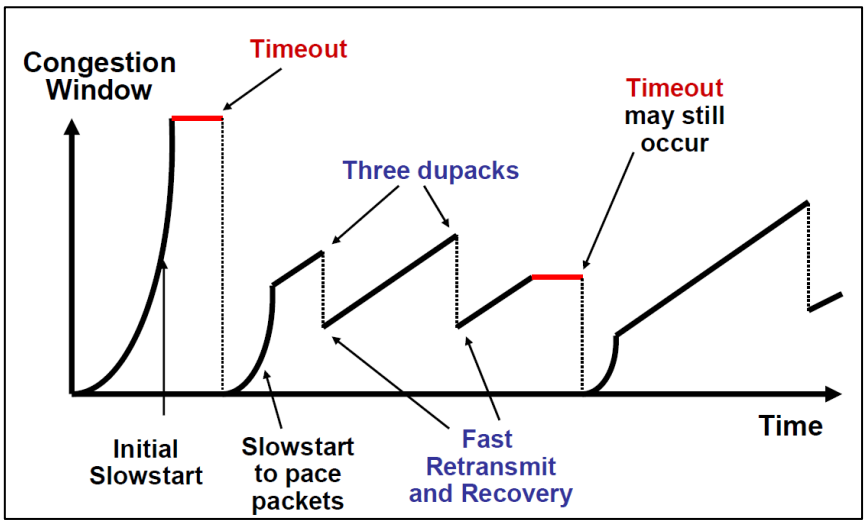
Until cwnd ≤ slow_start_threshold

- Slow Start phase (exponential growth)
  - Each returning ACK, a new pckt is transmitted
    - cwnd -> cwnd + 1
  - Every RTT
    - cwnd -> 2 cwnd

When cwnd > slow_start_threshold

- Congestion avoidance phase (linear growth)
  - Each returning ACK, a new pckt is transmitted
    - cwnd -> cwnd + (1/cwnd)
  - Every RTT
    - cwnd -> cwnd + 1

TCP sawtooth behaviour



TCP VEGAS

A version of TCP that tries to anticipate losses before they happen. Vegas does not use packet loss to predict congestion, but it uses **change in observed end-to-end delay to detect onset of congestion**. In fact, TCP Vegas implements a sort of congestion avoidance allowing to predict congestion before it happens (unlike Reno and New Reno). It compares the expected throughput to the actual one (Expected = window_size/RTT, Actual = acks/RTT). The actual throughput can be only equal or less than expected. The expected throughput is the transmission rate with no other traffic/queue.

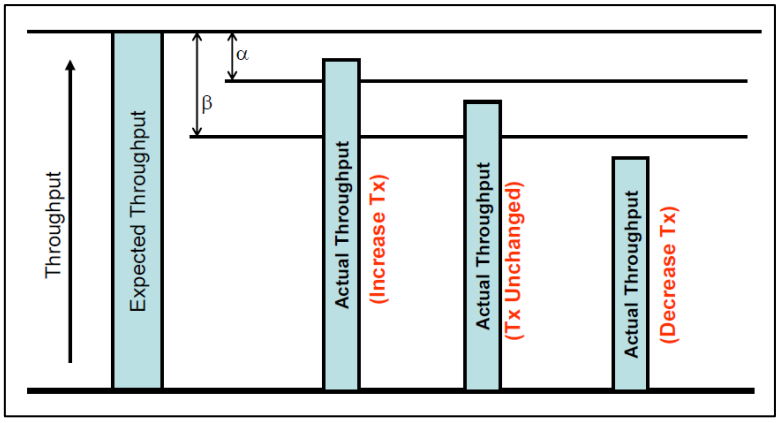
Monitor transmission rate (throughput, goodput):

- Thresholds of α and β correspond to how many packets Vegas is willing to have in queues
  - $\alpha < \beta$ so… expected - β < expected - α < expected
- If expected - α < actual < expected
  - Queues decreasing → increase rate
- If expected - β < actual < expected - α
  - Don't do anything
- If actual < expected - β
  - Queues increasing → decrease rate before packet drop

Vegas: Modified Congestion Avoidance

- Vegas Calculates (Once per RTT):
  - *Expected Throughput*=WindowSize/BaseRTT
  - *Actual Throughput*=ActualTransmittedAmount/RTT
- Static Parameters:
  - α = 1 pkts/RTT
  - β = 3 pkts/RTT

- TCP transmission rate = cwnd/RTT
- TCP takes cwnd updating decision once per RTT
- The decision is applied throughput the next RTT for each received ACK as follows:
  - Increase Tx Rate (Expected-Actual<α): :
    - cwnd = cwnd + 1/cwnd
  - Decrease Tx Rate (Expected-Actual>β):
    - cwnd = cwnd - 1/cwnd
  - Tx Rate Unchanged (α<Expected-Actual<β): :
    - cwnd = cwnd



Vegas: Aggressive Retransmission

- With dupacks
  - When Vegas receives the first dupack or the second dupacks, it checks the fine grained timer expiry
    - More aggressive retransmissions (helps in wireless environments with non-congestion losses)
  - If timer expirers, it retransmits immediately

Vegas: Aggressive cwnd Updating

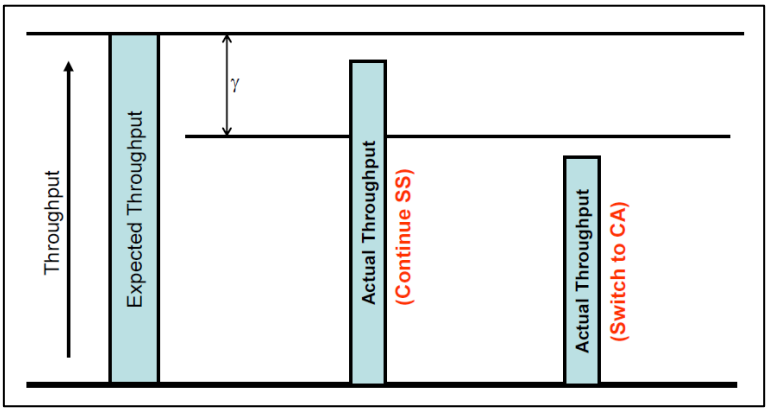
- With recovery
  - Reduce cwnd by one quarter instead of half when it enters into recovery
- With multiple loss
  - In case of multiple segment loss from a single window, it reduces the cwnd only once
- With Initial setting
  - cwnd is set to 2 instead of 1

More appropriate for error prone (wireless) environments

Vegas: Modified Slow-Start

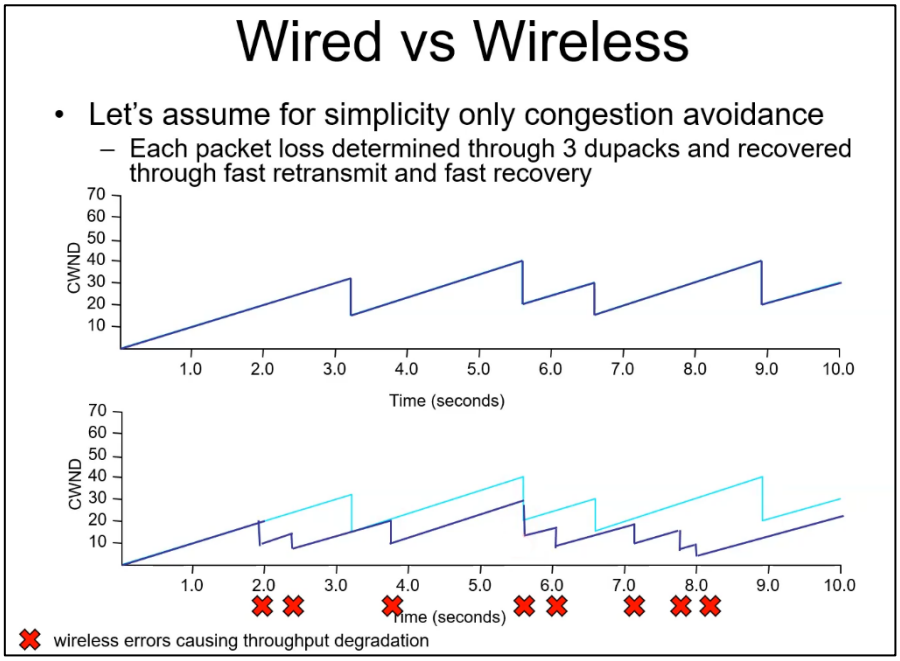
- **Vegas Calculates (in every alternate RTT):**
  - ❑ ***Expected Throughput*=WindowSize/BaseRTT**
  - ❑ ***Actual Throughput*=ActualSentAmount/RTT**
- **Static Parameters:**
  - ❑ **γ = 1 pkts/RTT**

- **TCP keeps the congestion window fixed in every other RTT and it measures the throughput**
- **On every next RTT, it does the followings:**
  - **Continue SS (Expected-Actual<γ):**
    - ❑ **Exponential Increase**
      - ❑ **cwnd = cwnd + 1 for each ACK, that is,**
      - ❑ **cwnd = 2 * cwnd for each RTT**
  - **Switch to CA (Expected-Actual>γ):**
    - ❑ **Set ssthresh=cwnd**
    - ❑ **Follow the rules of CA**



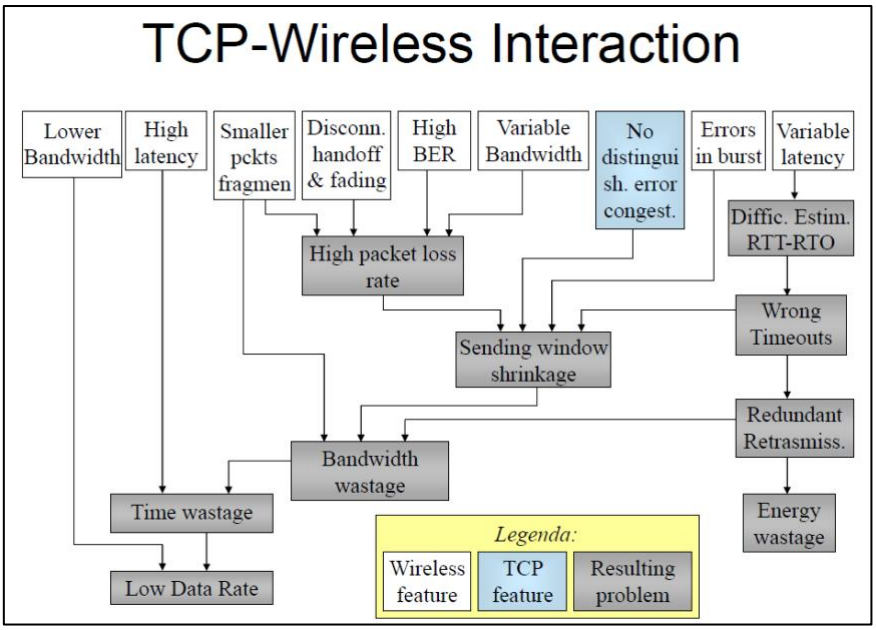
TCP Vegas is an interesting solution because it anticipates congestion, it allows to have a better exploitation of the channel without large ups/downs, all is managed smooth and linear.
As drawbacks it is sensible to delay variation which may be misinterpreted by the protocol, and **it cannot coexist with legacy TCP versions** (due to this never used). When a TCP Vegas flow shares a bottleneck link with TCP New Reno (for example), it detects congestion early and reduces its data rate as soon as the buffer starts filling. This creates additional space for TCP New Reno, which continues increasing its transmission rate until congestion occurs (packet loss due to buffer overflow).
Even if TCP Vegas has never been used, its ideas have constituted the basis of more recent TCP versions.

5.1 TCP over wireless



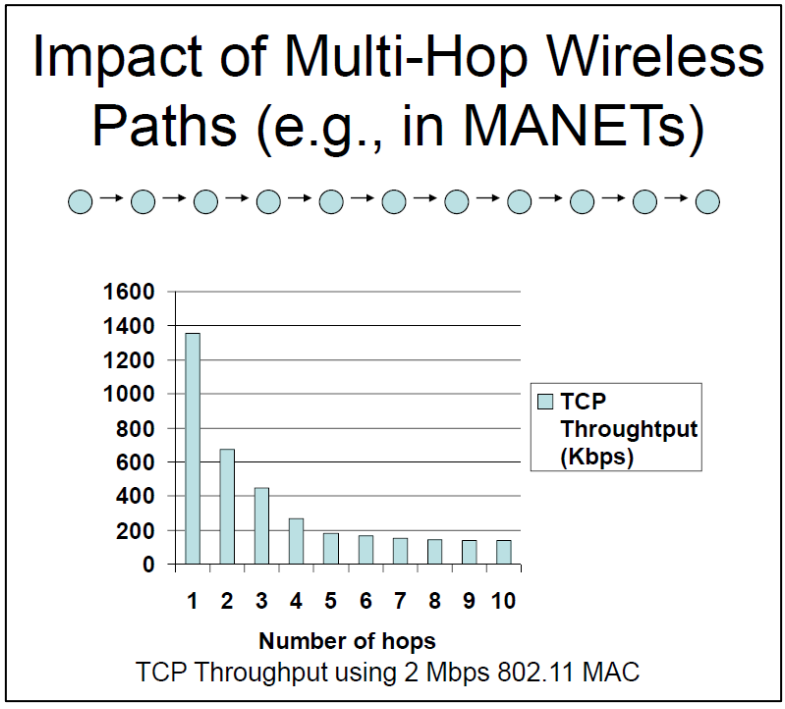
Problem: **every time there is a packet loss TCP slows down the transmission speed thinking about a possible congestion!** This means that every time we have interference in a wireless channel that cause a packet loss, the transmission speed is reduced leading to poor performances, even if there is no congestion (in that case we only want to retransmit the packet, but TCP does not reason in this way, as we have seen).
In the second plot of the above figure, it is possible to note the difference between the real points of congestion (following the first plot) and the points where TCP slows down the speed due to wireless errors, the difference between those two plots (wired – wireless) represents the total amount of wasted bandwidth.

As a result, standard TCP versions do not work efficiently with regular wireless links. In fact, TCP control mechanism was designed for a reliable medium, always treating losses as a sign of congestion. As a consequence, using standard versions of TCP on a wireless network result in several problem mainly due to the unique characteristics of wireless communications.

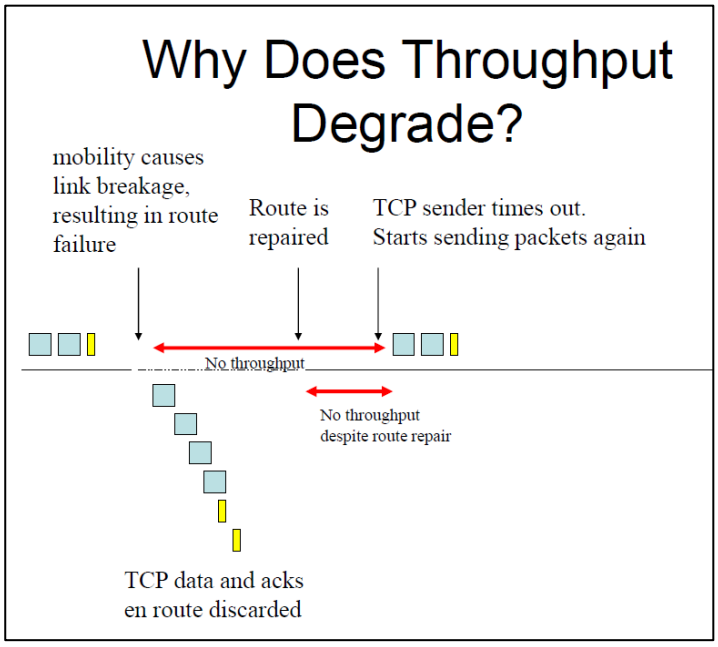


Wireless Problems: Impact on TCP

- **Error losses**: TCP assumes congestion and reduces cwnd.
- **Losses in bursts**: multiple cwnd reductions.
- **Long delays (satellites)**: RTT-unfairness. Considering the long delays due to satellite communications, RTT is higher and cwnd is increased in a much slower way, leading so to a reduced speed (even if the infrastructure allows us to do better!).
- **Variable delays**: wrong RTO computation.
- **Disconnections**: multiple timeouts.
- **Variable bandwidth**: sudden loss bursts or bandwidth wastage.

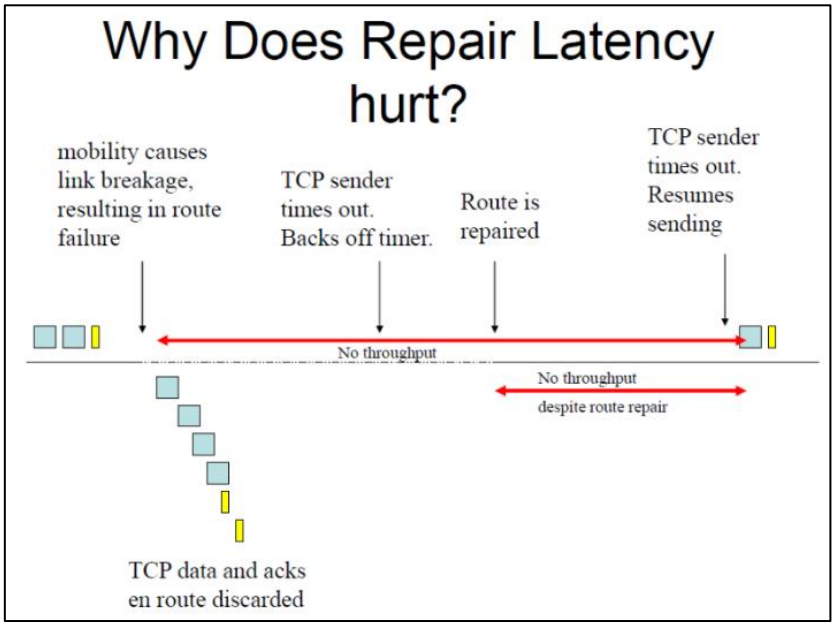


Consider multiple nodes connected one to another in a sort of "queue" where every node is in range with the previous and the following one. Now let's focus on the first two nodes, so considering a number of hops = 1. As we can see we reach a total throughput of almost 1400 Kbps (also considering there are other things that work in background as ACKs, time to wait for the channel, etc.). Now let's consider a number of hops = 2 where the first node of the sequence transmits the first message to node2 (hops=1) and then node2 forwards that message to node3. Anyway, suppose at the same time node1 wants to transmit another message to node2, node1→node2 and node2→node3 transmissions are in competition and they cannot transmit simultaneously, otherwise will result in a collision. If you apply this reasoning for the whole sequence the final throughput is highly reduced step by step, until the number of hops is large enough and the throughput stabilizes.



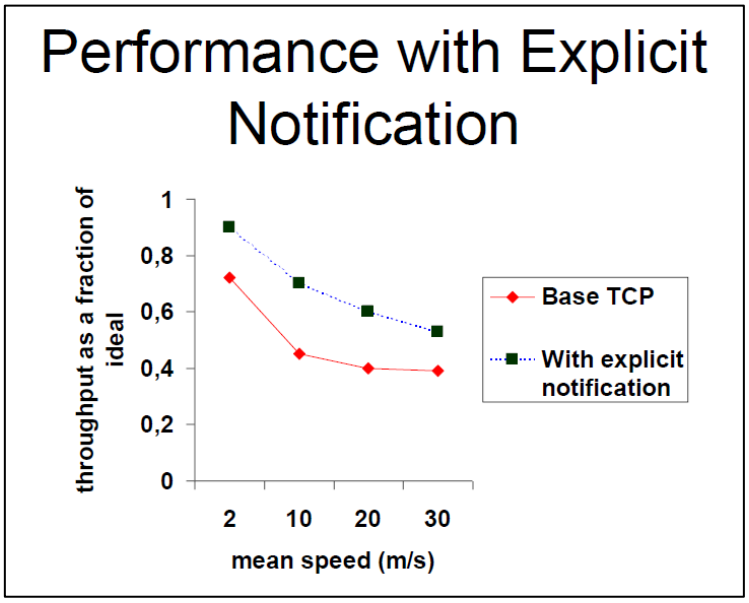
Throughput generally degrades with increasing speed (mobility)

When the link is broken due to mobility, we are not immediately conscious that "on-flight" packets were lost, we know that only once the timeout expires. So, we are not able to immediately retransmit the message once the route is repaired but we lose some time (bandwidth) waiting for the timer expiration, leading so to a poor exploitation of the channel.
The situation can be even worse if more timeouts are involved (look at next example).



Several improvements were proposed as solutions to alleviate problems generated by using standard TCP in wireless scenarios. In fact, the throughput may be improved, bringing it closer to the ideal, by:

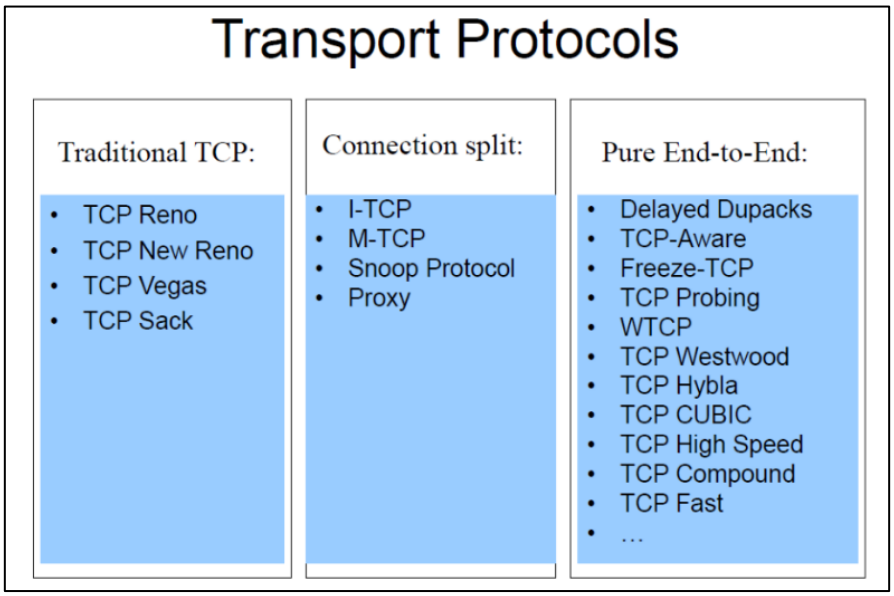
- **Network feedback**: using cross-layer network design is possible to improve performances by communicating the failure (e.g., link breakage in the previous examples) to the transport layer.
- **Inform TCP of route failure by explicit message**: it can be done in several ways, for instance creating an appropriate ACK or setting the advertised window to 0 in a way the transmitter selects the sending window as min{cwnd,0}, so 0, and stops sending packets, avoiding so the multiple timeouts expiration.
- **Let TCP know when route is repaired**: possible to do that using probing (sends small packets to see if the link is working or not) or explicit link repair notification.

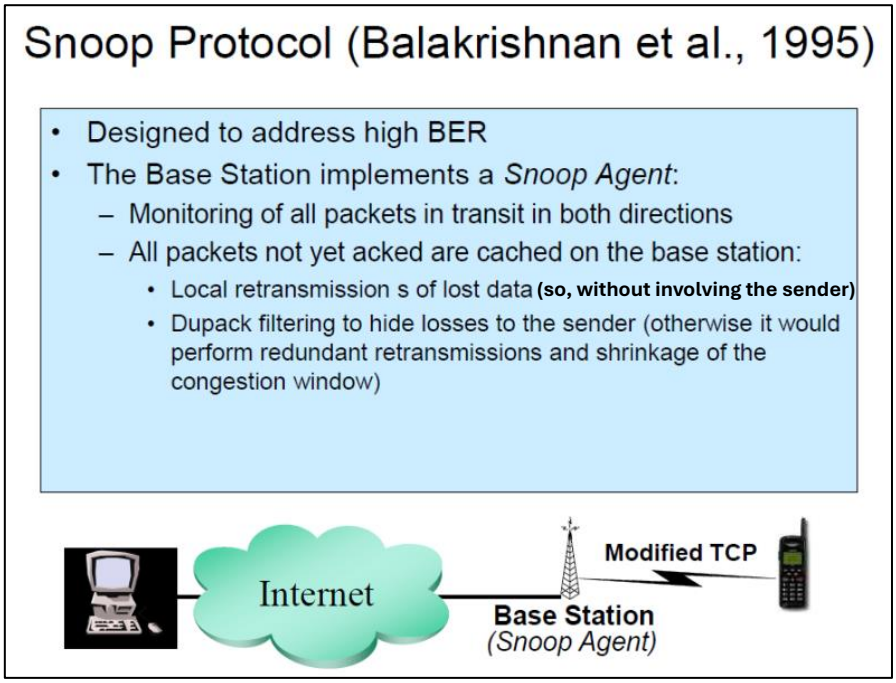


TRANSPORT LAYER SOLUTIONS

Connection split approach addresses wireless communication issues by introducing a proxy that splits a single connection into two segments (wired + wireless connections): for instance, one from the sender to an access point (AP) and another from the AP to the final receiver. Each segment can utilize different TCP protocols tailored to the specific network characteristics of each link, helping to mitigate wireless-related issues such as interference and signal loss.
However, this approach has a key drawback: **managing two separate connections can lead to a disconnect between the sender's perception of message delivery and the actual status of the transmission**. For example, when the sender transmits a message, it receives an acknowledgment (ACK) from the access point, leading it to believe the message has reached the receiver. In reality, the message may only have reached the AP, and any issue on the second segment (from AP to receiver) will remain unknown to the sender. Thus, the sender operates under the assumption of successful delivery, even if the message has not fully traversed the entire connection.

Pure End-to-End approach maintains a direct communication pathway between the sender and receiver, without splitting the connection at intermediate points. This approach is designed to preserve the integrity of the end-to-end communication paradigm, ensuring that data is transmitted directly from one endpoint to another. The pure end-to-end approach often **involves the use of a new or modified transport protocol that is specifically tailored to handle the challenges of wireless networks**. In this setup, the sender is aware of the wireless link characteristics, which allows it to adapt to potential issues like high error rates or variable latency inherent in wireless communication.





Snoop Protocol: Pros

- **End-to-End semantics preservation (almost)**: the base station (Snoop Agent) caches the ACKs created by the mobile station and that needs to be returned to the server (TCP sender). So, if an ACK is returned to the server this means the packet already reached the mobile station, avoiding so the previously discussed problem typical of connection split approach. Some dupack may be discarded because the snoop agent will recover the lost packet locally instead of triggering the sender again, this is not due to errors but to the nature of wireless transmissions. So, even if some dupack will be discarded this is not a big issue.
- **Local (and timely) loss recovery**.
- **Addresses high BER**.

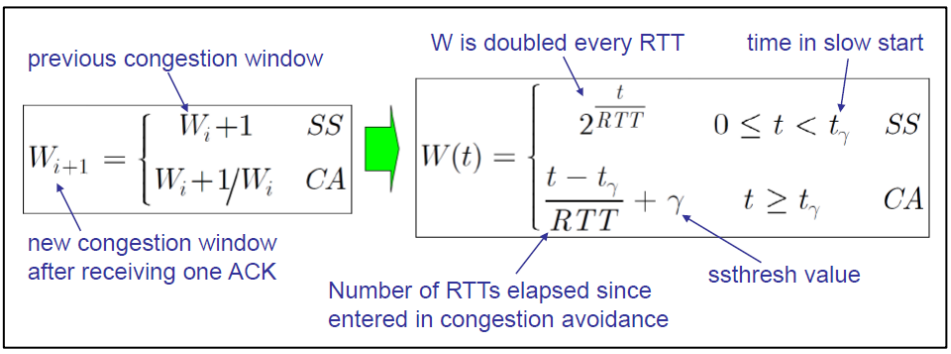
Snoop Protocol: Cons

- **Requires little RTTs on the wireless link**: the protocol performs best when the wireless link has a low RTT, as this minimizes the delay for detecting and retransmitting lost packets locally. If RTT is high on the wireless link, lost packets take longer to detect, and retransmissions are slower.
- **Does not guarantee against long disconnections**.
- **Not utilizable immediately after a handoff**: no packets in the new antenna's cache.

SATELLITE SCENARIOS

GEO satellites orbit at an altitude of about 36,000 km, while LEO satellites are much closer, ranging from 100 to 1,500 km above the Earth. There can be two possible configurations: *backbone configuration*, where satellites connect large networks or backbones, and *direct-to-home configuration*, where satellites provide a direct link to end-user terminals. A major challenge in satellite communication is the **high Round Trip Time (RTT)**, especially in GEO systems, which can reach up to 600 milliseconds due to the large distance signals must travel. Additionally, Packet Error Rate (PER) is a significant concern, as the radio channel is susceptible to interference, which can be caused by factors like satellite constellation layout, weather conditions, antenna alignment, and mobility. PER typically ranges between 0-10%, affecting the reliability and performance of satellite links.

Slow Start & Congestion Avoidance models



RTT unfairness

In satellite communication, the high RTT (such as up to 600 ms for GEO satellites) presents challenges for TCP because the **congestion window adjustments are based on RTT feedback**. With high RTTs, the Slow Start phase takes longer to ramp up, and packet losses can lead to significant reductions in *W*, impacting throughput. In synthesis, the longer the RTT the slower the *W* growth rate. Consider also the formula to compute the transmission rate B(t) (segments/sec) = *W(t) / RTT*, meaning that smaller the RTT is, higher is the final throughput.

TCP HYBLA

It was presented in 2004 with the aim of equalize the transmission rate against the RTT. They modified the original cwnd formula by adding a new parameter ρ = RTT/RTT_0, with RTT_0 as reference Round Trip Time (e.g., RTT_0 = 25ms). The main idea is to **take the reference speed of RTT_0 as the target one, even considering higher RTTs, by compensating**: for instance, when we receive an ACK, we should increase the cwnd by 2,3,4 packets and not just by 1.

$$W^H(t) = \begin{cases} \rho 2^{\rho \frac{t}{RTT}} & 0 \le t < t_{\gamma,0} \quad \text{SS} \\ \rho\left[\rho \frac{t - t_{\gamma,0}}{RTT} + \gamma\right] & t \ge t_{\gamma,0} \quad \text{CA} \end{cases}$$

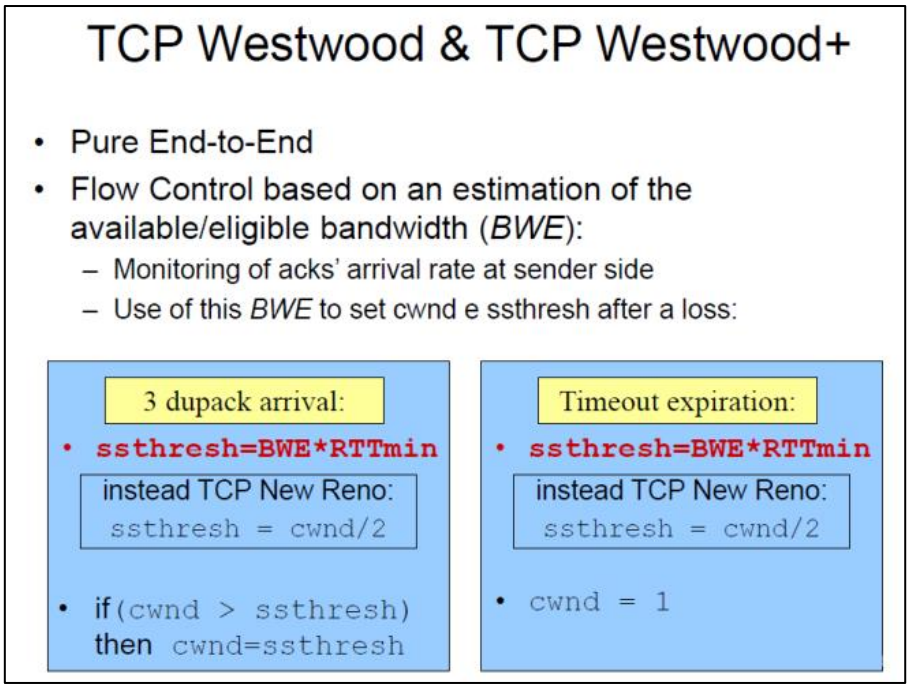
$$B^H(t) = \frac{W^H(t)}{RTT} = \begin{cases} \frac{2^{\frac{t}{RTT_0}}}{RTT_0} & 0 \le t < t_{\gamma,0} \quad \text{SS} \\ \frac{1}{RTT_0}\left[\frac{t - t_{\gamma,0}}{RTT_0} + \gamma\right] & t \ge t_{\gamma,0} \quad \text{CA} \end{cases}$$

The main advantage is that, regardless the real RTT value we can reach RTT_0 performances. In fact, in the above formula RTT_0 is constantly reported, imposing it as our target.

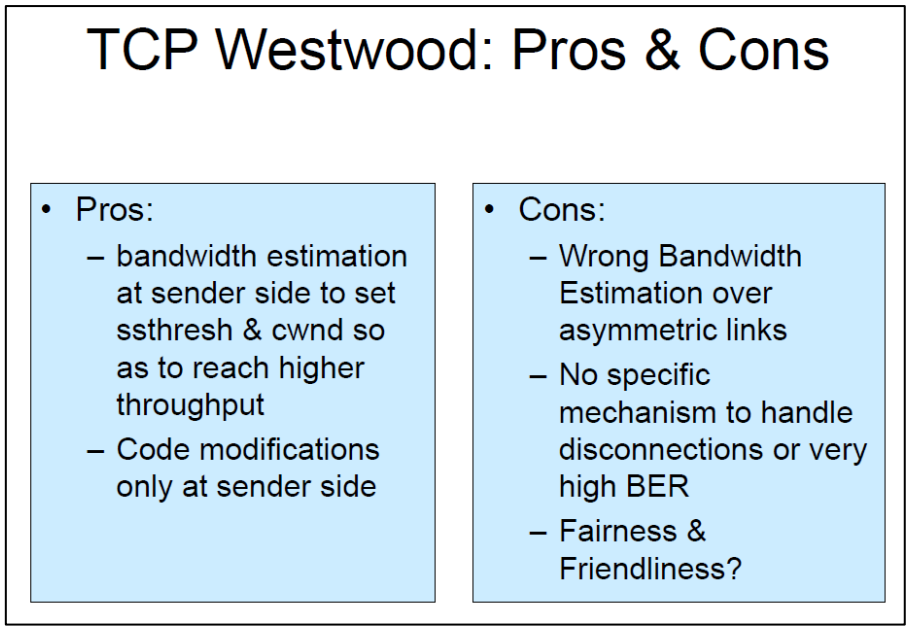
TCP Hybla: Pros & Cons

- Pros:
  - End-to-End solution
  - Code modifications only at sender side
  - RTT used to speed up transmission speed for connections with long RTTs (e.g., satellites) in order to reach RTT fairness
- Cons:
  - Agressive behavior may result in multiple losses
  - Measured RTT is sensitive to buffer size*
  - No handling of BER or disconnections
  - Fairness & Friendliness?

* If network buffers are too large, packets may be queued for a longer time before being transmitted, artificially inflating the RTT measurement.



While transmissions keep going, the system tries to monitor the performances to determine the speed that worked best, without delaying, losses and so on. When there is a packet loss the idea is to bring the current bandwidth to the value that "in the past" worked best. The only difference from what we know so far is the way ssthresh is computed to enhance congestion control. The sender computes *rate estimation* (*RE*) by sampling and exponential filtering during the interval T (typically the RTT). Samples are determined from ACK inter-arrival times and info in ACKs regarding amounts of bytes delivered. The RE is then used to properly set cwnd and ssthresh after packet loss (indicated by 3 dupacks or timeout).



TCP Adaptive-Selection

- Going back to end-to-end enhancements, few question arise:
  - is it necessary to make a definitive choice among TCP enhancements?
  - Why not to select optimized TCP variant on different connections on the same server in an adaptive way?
  - Is there any room for performance improvement?
  - Is it feasible to simultaneously run different TCP enhancements on the same machine?
- The Adaptive-Selection* concept try to answer to all these questions

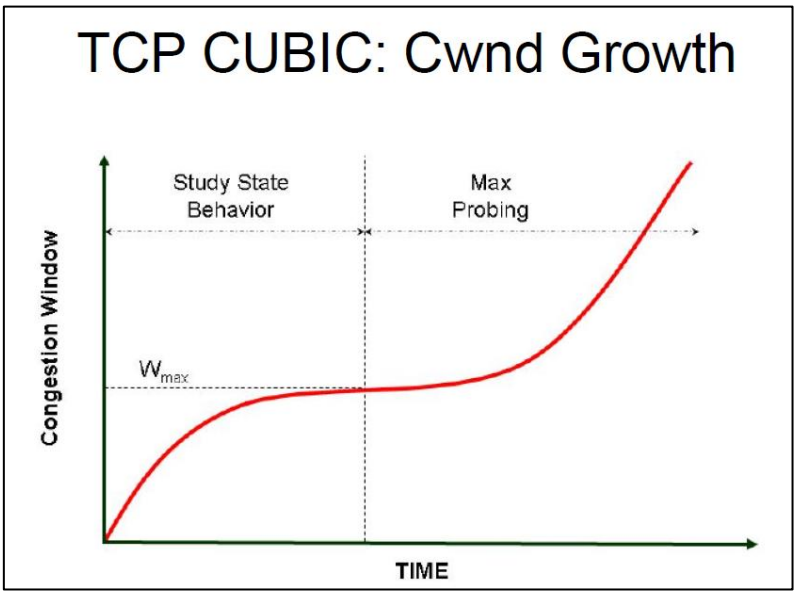
- The TCP adaptive-selection concept is very simple:
  - On the same server not a single TCP variant, but concurrent use of different TCP enhancements to match the different characteristics of connections.
- It can be applied in different ways, depending on:
  - the agent that performs the TCP selection (i.e. receiver, intermediate router, sender)
  - the possible exploitation of a cross-layer approach
  - the possibility to change the TCP version on an on-going connection
    - "dynamic" adaptive-selection (like gears in a car)

- Linux OS appears the most convenient choice to implement TCP adaptive-selection
  - Most TCP variants are already available as modules
  - A new "TCP adaptive-selection" module that calls other modules could be the solution
- Several possibilities for the decision criteria
  - TCP internal parameters (such as RTT and /or Bandwidth estimation)
  - Cross-layer information
  - Reliable channel estimation

Need for quick and efficient metrics to determine best choice at any time

TCP CUBIC

It represents a significant advancement in congestion control algorithms, specifically designed to optimize performance in high-speed networks with high latency, commonly known as Long Fat Pipes/Networks. Its distinctive feature lies in its congestion window management, which employs **a cubic function based on the time elapsed since the last congestion event**. The algorithm initializes with rapid growth, then intelligently adjusts by slowing down and maintaining stability around the window size where congestion previously occurred. If network conditions improve (such as when other flows exit the network), TCP CUBIC quickly resumes growth to utilize the newly available bandwidth. What sets it apart from traditional TCP implementations is that it does not depend on ACK receipts for window size increases; instead, it bases its decisions solely on the last congestion event. This approach **significantly reduces RTT-unfairness since window growth is independent from RTT**. Due to its effectiveness, TCP CUBIC has become the default congestion control algorithm in Linux kernels version 2.6.19 and above.



6. IEEE 802.11 standards

The standard IEEE 802.11 has been constantly updated during the time, following more data demand as video streaming, etc. Here below you can find the different versions.

Standard

Gruppi di standardizzazione IEEE 802.11	Descrizione
IEEE 802.11	lo standard originale: bitrate da 1 a 2 Mbps, spettro 2.4 Ghz, livello fisico sia radio che infrarosso
IEEE 802.11a	54 Mbit/s, 5 GHz, lanciato nel 2001
IEEE 802.11b	sviluppo di IEEE 802.11 (1999), da 5.5 a 11 Mbps
IEEE 802.11d	estensioni per roaming internazionale
IEEE 802.11e	estensioni per qualità del servizio
IEEE 802.11f	standard per Inter Access Point Protocol (IAPP[2])
IEEE 802.11g	54 Mbit/s, 2.4 GHz, retrocompatibile con IEEE 802.11b
IEEE 802.11h	selezione dinamica dei canali e controllo della potenza trasmissiva (compatibile con direttive europee)
IEEE 802.11i	integrazioni e estensioni per la sicurezza (2004)
IEEE 802.11j	estensioni per direttive giapponesi
IEEE 802.11k	estensioni per misurazione dei parametri radio
IEEE 802.11n	estensioni per throughput elevati (oltre 200 Mbps) mediante tecnologia MIMO (trasmettitori e ricevitori multipli)
IEEE 802.11p	accesso wireless per sistemi veicolari (WAVE)
IEEE 802.11r	estensioni per roaming veloce
IEEE 802.11s	estensioni per reti wireless mesh
IEEE 802.11t	metodi e metriche per misurazione e predizione delle prestazioni
IEEE 802.11u	internetworking con reti non 802.11 (cellulari)
IEEE 802.11v	gestione e amministrazione delle reti wireless

Recent Standards

- IEEE 802.11n (Wi-Fi 4 – new denomination by Wi-Fi Alliance)
- IEEE 802.11ac (Wi-Fi 5)
  - Expanded use of MIMO (up to 8) and wider band (160MHz) than 802.11n
  - Up to 500 Mbps for each single connection
- IEEE 802.11ax (Wi-Fi 6)
  - Frequency: between 1 and 7 GHz (and not only 2.4 and 5 GHz)
  - Up to 11 Gbps in test
  - Lower latency (less than half with respect to 802.11n)
  - Modified MIMO: UL MU-MIMO (uplink multiuser multiple-input multiple-output)
- IEEE 802.11be (Wi-Fi 7)
  - Frequency: 2.4, 5 and 6 GHz
  - Modified MIMO: CMU-MIMO (coordinated multiuser MIMO)
- IEEE 802.11mc
  - Indoor localization (1-2 m precision)
    - using Wi-Fi Round Trip Time (Wi-Fi RTT)

6.1 IEEE 802.11e

The IEEE 802.11e standard is an enhancement to the original IEEE 802.11 standard, focusing on improving Quality of Service for Wi-Fi networks. This standard did not meet the market, but it has been used as base for IEEE 802.11p for vehicular networks. In general, it is important to know how it works.

Originally IEEE 802.11 standard defined exchanges using DCF and PCF. IEEE 802.11e introduced new mechanisms for QoS through priority and parameterized QoS schemes.

- **PRIORITY SCHEME: EDCA** – Enhanced DCF Channel Access (WMM Wi-Fi Multi Media)
- **PARAMETRIZED QOS SCHEME: HCCA** – Hybrid Coordinator Function Channel Access (WMM-SA Wi-Fi Multi Media Scheduled Access)

DCF is the basic access method in IEEE 802.11 networks, using CSMA/CA to manage channel access. DCF first checks for activity on the medium; if it is free, the device waits for a DIFS plus a contention period, then transmits if the medium is still clear. If the medium is busy, the device initiates a random back-off process, selecting a number of slots between 31 and 1023. The device counts down these slots while the medium remains idle, and when the countdown reaches zero, it attempts to transmit. If a collision occurs, the device backs off again, increasing the random window size up to a preconfigured maximum.

802.11 Contention Window

The 802.11 Contention Window (CW) is the range from which a random back-off number is chosen to manage access to the medium. A random number is selected from within the range [0,CW]. Using a smaller CW value can lead to fewer wasted idle slots but increases the likelihood of collisions when multiple devices reach zero simultaneously. An optimal CW value can be calculated when the number of contenders and packet size are known, minimizing time wasted from collisions and empty slots. However, **implementing an optimal CW is challenging** because the number of active contenders is often dynamic and difficult to estimate accurately. We need something that is adaptable over time.

802.11 Adaptive Contention Window

The 802.11 Adaptive Contention Window dynamically adjusts the CW based on transmission success. It starts with a CW of 31; if no Clear to Send (CTS) or ACK is received, the CW is doubled and incremented by one (e.g., 63, 127, 255). Upon a successful transmission, the CW resets to 31. However, this adaptive scheme can be **unfair in high contention environments**: unlucky nodes may face a larger CW than lucky nodes, which quickly reset after successful transmissions. This can result in lucky nodes transmitting multiple packets while unlucky nodes wait to access the channel. Additionally, the adaptive CW mechanism does not provide Quality of Service (QoS).

PCF is a priority system that operates under centralized control. The PC (Point Coordinator), which typically also serves as the Access Point, manages this function. PCF operates with two key periods: CP (Contention Period) and CFP (Contention Free Period), which occur after each

Beacon. The system employs PIFS to maintain control, which is shorter than any DCF. Additionally, the PC maintains a list of stations that are eligible for polling.
PCF comes with several drawbacks. The function is fixed to a specific length of time after a Beacon and must synchronize with Beacon intervals. This makes it incompatible with voice or video streams that require intervals of 10, 20, or 30ms. Furthermore, PCF lacks any mechanism to reserve bandwidth or characterize network traffic, and it cannot handle back-to-back packets. Due to that, PCF is not used in practice.

Enhanced DCF Channel Access (EDCA)

- It introduces 4 Access Categories (AC) with 8 Traffic Classes (TC), a sort of metric.
- MSDU (Max Service Data Unit – what we want to transmit) are delivered through multiple back offs within one station using AC specific parameters.
- Each AC independently starts a back off after detecting the channel being idle for AIFS.
- After waiting AIFS, each back off sets counter from number drawn from interval [1,CW+1].
- **newCW [AC] ≥ ((oldCW [TC] + 1) * PF)**

The Persistence Factor (PF) is used to adjust the CW dynamically, influencing how aggressively or conservatively a station attempts retransmissions after a failed transmission. The value of PF depends on the AC and is used to control the exponential increase of the CW.

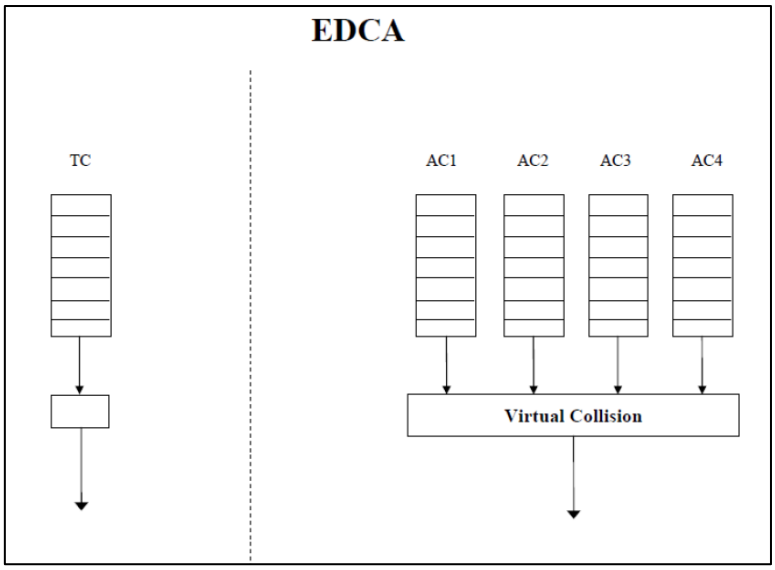
Consider the following example to better understand how EDCA works having 4 different AC, such as voice, video, best effort and background with different importance.

	AC_VO [0] (voice)	AC_VI [1] (video)	AC_BE [2] (best effort)	AC_BK [3] (background)
AIFSN	2	2	3	7
CWmin	3	7	15	15
CWmax	7	15	1023	1023

Prioritized Channel Access is realized with the QoS parameters per TC, which include :

- AIFS[AC] (Arbitration Inter-frame Space)
- CWmin[AC]
- PF[AC] (Persistence Factor)

The idea is to parametrize each AC in a way to reach QoS. So, different AIFS, CW and PF depending on the class.



On the left side, we can see TC (Traditional Channel Access) which shows a single queue structure where all traffic types are handled in the same way, regardless of their priority or requirements. This represents the basic DCF approach used in IEEE 802.11.

On the right side instead, we can observe the EDCA structure which introduces four separate Access Categories (AC1, AC2, AC3, and AC4). Each AC represents a different priority queue for different types of traffic. These queues feed into a "Virtual Collision" handler. The virtual collision mechanism resolves internal conflicts when multiple ACs have packets ready to transmit at the same time, typically giving preference to higher-priority queues. Each AC operates with its own set of parameters for channel access, making the system more flexible and capable of handling different traffic requirements more effectively than the traditional single-queue approach.

EDCA – Priority Scheme

EDCA is effectively DCF with 4 priorities.

User Priority 802.1D	Access Class	Designation
1 & 2	AC0	Background
0 & 3	AC1	Best Effort
4 & 5	AC2	Video
6 & 7	AC3	Voice

↓ Highest Priority

Bursting is possible: AC2 (AC_VI) TXOP limit 3ms

AC3 (AC_VO) TXOP limit 1.5ms

By setting different min and max back-off slots, one stream has an advantage over another. These max & min back-offs are configurable through the management interface, but choosing the optimum values for every scenario is not obvious

The main problem of this approach is how to determine to which category a flow belongs to. There should be something that marks the flow.

Example :

	AC[0]	AC[1]	AC[2]
AIFSN	2	4	7
CWmin	7	10	15
CWmax	7	31	255
PF	1	2	2

AIFS[AC] = AIFSN[AC] * aSlotTime + SIFS

PIFS - 25 µsec (*Used in HCCA*)
SIFS - 16 µsec
Slot Time - 9 µsec

AIFS[0] = (2 * 9) + 16 = 34 µsec = DIFS
AIFS[1] = (4 * 9) + 16 = 52 µsec → (52 – 34) / 9 = 18/9 = 2 Slots
AIFS[2] = (7 * 9) + 16 = 79 µsec → (79 – 34) / 9 = 45/9 = 5 Slots

Back-off Algorithm :

802.11 : $CW_{RANGE} = [\,0\,,\,2^{2+i} - 1\,]$

802.11e : newCW[AC] = [(oldCW[AC] + 1) * PF] - 1

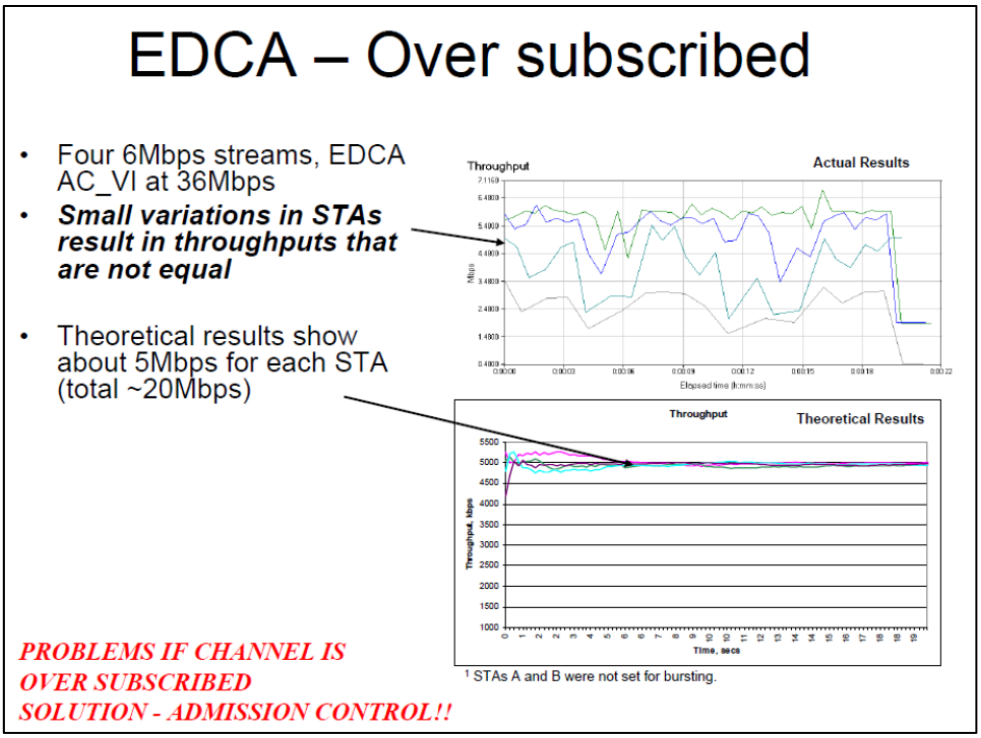
	Collision1	Collision2	Collision3
AC[0]	[(7+1)*1]-1 = 7 (0 - 7)	(0-7)	(0-7)
AC[1]	[(10+1)*2]-1 = 21 (0 - 21)	[(21+1)*2]-1 = 43 (0 – 31)	(0 – 31)
AC[2]	[(15+1)*2]-1 = 31 (0 – 31)	[(31+1)*2]-1 = 63 (0 – 63)	[(63+1)*2]-1 = 127 (0 – 127)

Advantages of EDCA

- Voice and Video streams have priority over data.
- Works well if network is lightly loaded, such as a voice-based network.
- No stream set-up instructions required.
- EDCA Power Save is big advantage over legacy power save.

Drawbacks of EDCA

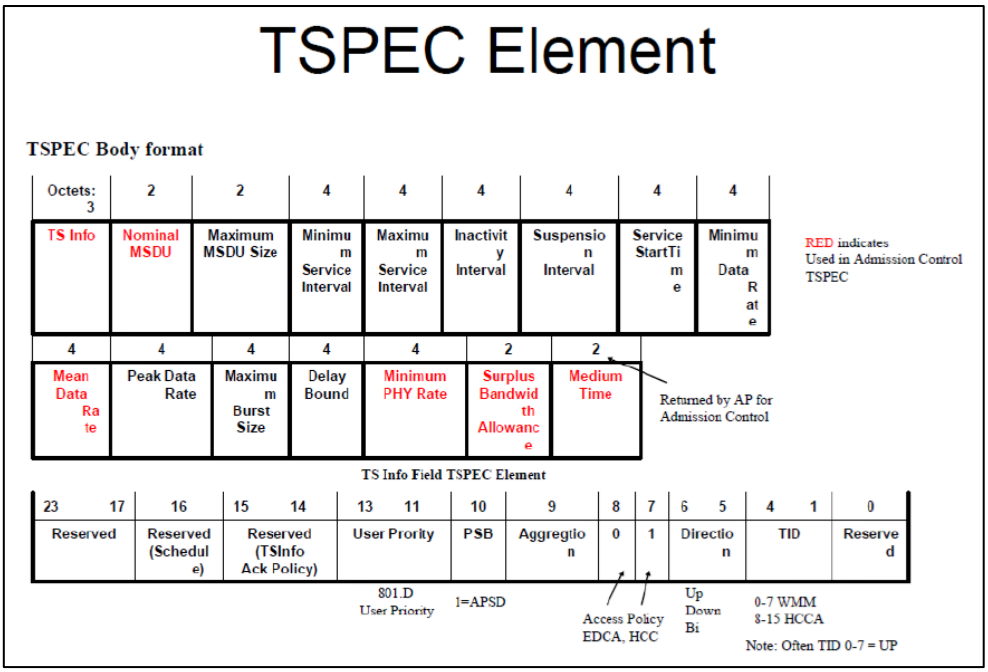
- Still based on "Fairness", lower priority must still get through.
- Streams of the same priority compete; not able to guarantee access, Latency, BW or Jitter.
- EDCA relies on every individual STA and the AP to control the priorities and access to the medium. Variations in QoS performance do occur in practice.
- Admission Control is used to overcome some of these disadvantages.



ECDA/WMM has no guarantees for QoS, but Admission Control can be used to improve situation: **limit admission to an Access Category** (e.g., voice and video). Without admission control, many high-priority traffic streams may enter the network, leading to excessive collisions and delays.

Admission Control

- AP advertises ACM bit in Beacon to indicate if admission control is mandatory for any Access Category.
- To use AC that has ACM bit set, STA (station) sends AddTS (Add Traffic Specification) Request Action Frame to AP that includes a TSPEC.
    - Nominal MSDU size
    - Mean Data Rate
    - Min PHY Rate
    - Surplus Bandwidth Allowance (SBA)
- AP runs the admission control algorithm and communicates back to the station the admission decision using AddTS Response Action frame.
    - Medium Time
- STA checks "Used Time" over 1 second periods.
    - If Used Time > Medium Time, STA must cease using that AC's EDCA parameters.



Admission Control represents an improvement over EDCA/WMM as it aims to manage higher priority streams and provide protection for streams that are already in progress. The system requires TSPEC configuration, which involves inputting the basic parameters of the QoS stream, with the station being responsible for sending the TSPEC. However, a notable limitation exists - since streams continue to contend with each other, the bandwidth efficiency does not achieve optimal levels.

Hybrid Coordinator Function Channel Access (HCCA)

HCCA is extension of PCF, uses Contention Free Periods (CFP). Hybrid Controller (HC) can initiate HCCA, CFP:

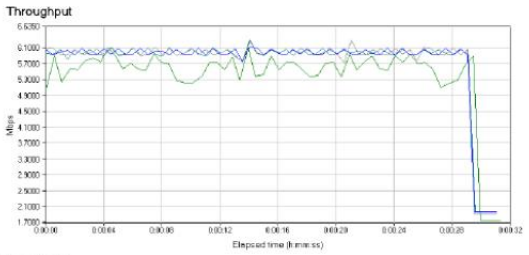
- Provides CF-Poll to station to provide TXOP (transmission opportunity).
- Specifies start time and maximum duration (hence other stations do not attempt to access the medium).
- Station (STA) transmits within SIFS and then using PIFS periods.
- If no transmission after a PIFS, HC takes over and issues new TXOP or end CFP.
- CFPs can be synchronized to the individual source traffic intervals instead of the Beacon intervals.

The HCCA introduces significant improvements over the traditional PCF system in terms of timing flexibility and access control. While PCF strictly divides the interval between beacon frames into Contention Free Period and Contention Period, HCCA implements a more dynamic approach by allowing CFPs to be initiated at virtually any time during a CP through what is known as a **Controlled Access Phase (CAP)**. The AP has the authority to initiate a CAP whenever frame transmission or reception needs to occur in a contention-free environment. During these CAP periods, the Hybrid Coordinator (HC), which is essentially the AP, maintains full control over medium access. Outside of CAP, during the CP, stations operate using EDCA protocols. A notable enhancement of HCCA is that the HC's capabilities extend beyond simple

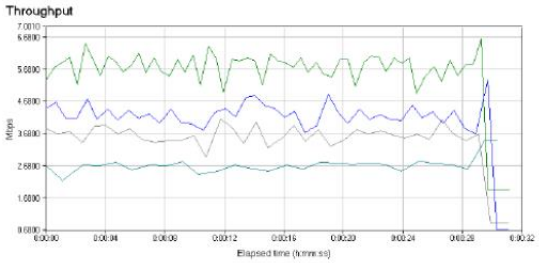
per-station queuing, offering more sophisticated per-session service with flexible stream coordination patterns, moving beyond the basic round-robin approach of PCF.

HCCA Efficiency - Measurement

As HCCA uses contention free periods to send the streams, the bandwidth efficiency is good. Examples below show that the practical difference.



WMM-SA
Four 6Mbps up-streams at 36Mbps
3 using WMM-SA
- **~24Mbps** throughput



WMM
Four 6Mbps up-streams on 4 different WMM certified devices. WMM AC_VI. STAs connected at 36Mbps
- **~16.5Mbps** throughput

HCCA - Summary

- Efficient use of Bandwidth
  - Contention free periods used
  - Returns channel as soon as packets sent for that TXOP
- "Guarantees" latency
  - Important in high bandwidth streaming applications
  - Regularly grants TXOPs as required by the TSPEC
- "Guarantees" Bandwidth
  - For quality video stream, for example, data rate must be assured
  - Very efficient use of available bandwidth, e.g. # of simultaneous voice calls is much higher than WMM allows (due to limited back-off slots)
- Overcomes most OBSS problems
  - All STAs and APs that hear the QoS Poll will obey the TXOP
  - ACKs from QSTAs should include Duration Field with outstanding TXOP time – extends range of CFP to other networks
  - Overlapping HCCA networks do have TXOP problems, this is being solved in 802.11aa

BUT

- Requires a complex Scheduler and added complexity

QoS requirements

The 802.11 QoS framework comprises two main approaches: EDCA Admission Control and HCCA. Both mechanisms rely on Traffic Specifications (TSPECs) to function effectively. For implementing either approach, certain fundamental parameters of the QoS stream must be known and specified in the TSPECs, including the Nominal MSDU size and Mean Data Rate. Additionally, HCCA has a specific requirement for Maximum Service Interval parameter. The HC (for HCCA) or the AP (for EDCA with admission control) uses TSPECs to decide whether a new traffic stream can be accommodated without affecting existing QoS.

6.2 IEEE 802.11n

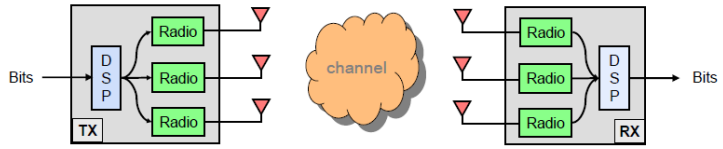
Even if nowadays this standard has been replaced by newer ones, it is interesting to study how it works because it was the first to introduce **MIMO (Multiple Input Multiple Output)** technology. MIMO allows to significantly increase the total amount of bandwidth without changing the assigned frequency, making suitable 802.11n for larger applications such as enterprise applications, video streaming and so on. High data rates between 64 and 600 Mbps. Using two transmitter MIMO device will support a 300 Mbps data rate when using 40 MHz channel (144 Mbps when using 20 MHz channel).

IEEE 802.11n introduces several enhancements to both the Physical Layer and MAC layers of the Wi-Fi standard. These enhancements aim to improve data rates and overall network performance.
On the PHY layer, OFDM modulation is supported with additional coding methods, preambles, multiple streams, and beamforming. MIMO technology with spatial multiplexing is also introduced. Additionally, high-throughput PHY with 40 MHz channels is supported, enabling higher data rates.
On the MAC layer, two MAC aggregation methods are supported to efficiently pack smaller packets into a single MPDU. Block Acknowledgement is another feature that allows for combined acknowledgements to be sent at a later point in time, optimizing network efficiency.

802.11n allows to pack more packets (data and ACKs) into only one. This is convenient when the channel is unreliable since, in that case, sending more packets will results in high losses.
Also consider that sending out less packets allows to reduce the total amount of “wasted” time for channel access. That’s why in some cases the bandwidth is not fully exploited. The idea is trying to find a balance between the number of packets and the wasted time for channel access.

What is MIMO?

- Multiple Input Multiple Output (MIMO)
  - Transmit and Receive with multiple radios simultaneously in same spectrum



- Compare to traditional Single Input Single Output Radio (with optional receive diversity)

Conventional (SISO) Wireless Systems

Conventional "Single Input Single Output" (SISO) systems were favored for simplicity and low-cost but have some shortcomings:

- Outage occurs if antennas fall into null
  - Switching between different antennas can help
- Energy is wasted by sending in all directions
  - Can cause additional interference to others
- Sensitive to interference from all directions
- Output power limited by single power amplifier

MIMO Wireless Systems

Multiple Input Multiple Output (MIMO) systems with multiple parallel radios improve the following:

- Outages reduced by using information from multiple antennas
- Transmit power can be increased via multiple power amplifiers
- Higher throughputs possible
- Transmit and receive interference limited by some techniques

7. Vehicular Ad-Hoc Networks (VANETs)

Vehicular Ad Hoc Networks (VANETs) represent a specialized subset of Mobile Ad-Hoc Networks (MANETs) designed to enable communication among vehicles and between vehicles and roadside infrastructure. VANETs are necessary for enhancing road safety, improving traffic efficiency, and enabling diverse applications such as infotainment and navigation.

7.1 IEEE 802.11p

IEEE 802.11p is a dedicated protocol for vehicular communication, operating within the broader family of IEEE 802.11 standards. It extends connectivity to vehicles, opening the way for applications that range from public safety to traffic management.

IEEE 802.11p utilizes a reserved frequency band at 5.9 GHz, providing a transmission range of up to 1000 meters under ideal conditions. At closer distances, it supports data rates of up to 26 Mbps. However, at a range of 300 meters, this rate may drop to 6 Mbps, still sufficient for exchanging critical messages between vehicles traveling at speeds up to 200 km/h.
Furthermore, it incorporates elements from earlier IEEE 802.11a and 802.11b standards, optimizing them for the high-mobility environment of vehicular networks.

One primary challenge is **ensuring seamless connection between Access Points (APs) and fast-moving vehicles**. As transmission ranges change dynamically with vehicle speed and position, maintaining synchronization becomes critical. Variability in transmission power and range can lead to delays or interruptions, especially in high-mobility scenarios.

7.2 System model

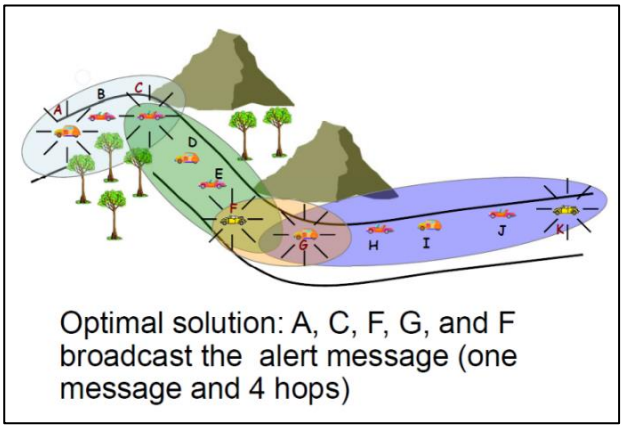
The system model for VANETs is tailored to address key requirements such as safe driving and rapid alert dissemination. The goal is to enable **efficient broadcasting of alert messages to minimize accidents**, particularly in scenarios involving abnormal vehicle behaviour.

Core requirements:

- **Alert propagation**: vehicles need to broadcast alert messages that trigger a chain reaction among following vehicles.
- **Challenges**:
    - Multiple transmissions: concurrent broadcasts by multiple vehicles can lead to congestion.
    - Congestion: excessive communication may render alerts ineffective, increasing the risk of accidents.

Key characteristics:

- **High node mobility**: vehicle speeds and positions change constantly, necessitating adaptive communication.
- **Variable transmission ranges**: signal range fluctuates based on vehicle location, traffic density, and environmental conditions.
- **Uncertainty of coverage**: a car cannot be sure to be the farthest car receiving that broadcast message.



Optimal solution: A, C, F, G, and F broadcast the alert message (one message and 4 hops)

Who has to forward the alert message?

How can C know that is the farthest vehicle in range of A? Not so simple to determine. Hello messages can be used to perceive the closer vehicles but cannot be used to solve the problem of coverage.

Different approaches to solve the problem of uncertain coverage:

- **Minimum Connected Dominant Set (MCDS)**:
  - MCDS notion: the minimum cardinality set of connected nodes, such that each other node in the network is connected to a node of the MCDS set.
  - Defines a minimal set of connected nodes responsible for broadcasting.
  - Advantages: optimal coverage with reduced redundancy. Anyway, MCDS is not a feasible solution since it requires a complete and updated knowledge of the network topology.
  - Drawbacks: prone to deterministic failure if a node in the MCDS fails to broadcast (dangerous scenarios). Due to that, it requires redundancy and control messages with a complexity of $O(n \cdot \log(n))$, considering n nodes in the network.
- **Redundancy Avoidance approaches**:
  - Uses a backoff mechanism to prevent collisions due to congestion.
  - If the message has already been rebroadcast by the following vehicle do not forward it (it would be redundant).
  - Limitation: does not account for the number of hops that a message traverses.
- **Jamming Signal**:
  - Employs a jamming signal to determine the next forwarder.
  - Vehicles receiving an alert message emit a jamming signal for a time that is proportional to the distance from the sender.
  - The last vehicle, knowing it is the last one (since it listens the channel to discover other jamming signals), stops the jamming signal and forwards the message.
  - Limitation: introduces delays in the transmission of the message, this makes this solution unsuitable for time-sensitive alerts.

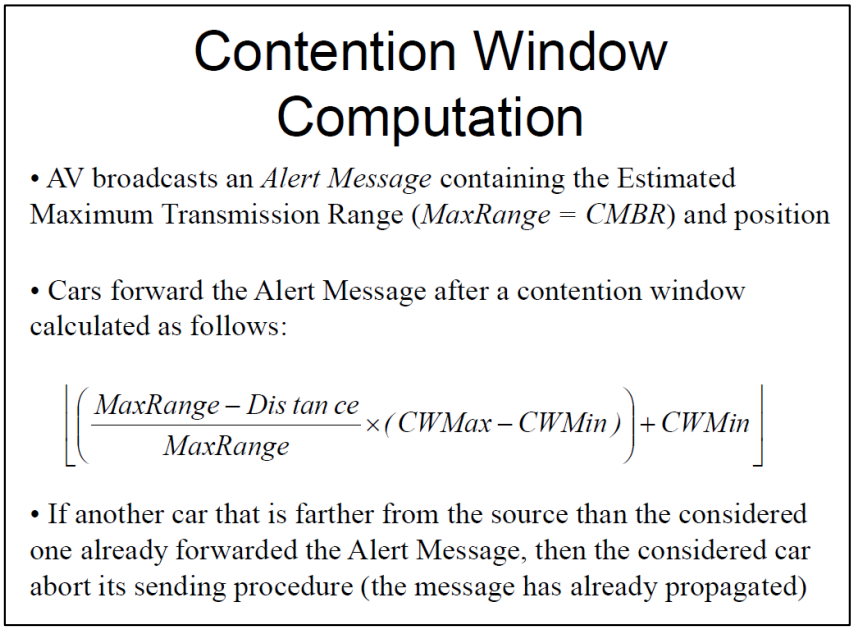
- **Contention Window**:
  - Sets the contention window inversely proportional to the distance from the sender, meaning that further nodes are faster.
  - Advantages: eliminates control traffic.
  - Drawbacks: unrealistic assumptions about uniform transmission rates.

7.3 Fast broadcasting

Fast broadcasting schemes aim to disseminate alert messages across an area of interest with minimal delay (so, using the minimum number of hops), ensuring timely communication in emergencies.

Fast broadcasting works in two operational phases:

- **Estimation Phase**:
  - Vehicles exchange hello messages to gather positional and range information.
  - Each hello message contains:
    - Sender's position.
    - Maximum forward distance from which the sender received another hello message.
  - CMBR (Current Maximum Backward Range): the farthest range at which the sender's message is heard by a vehicle behind.
  - CMFR (Current Maximum Forward Range): the farthest range from which the sender receives a hello from a vehicle ahead.
- **Broadcast Phase**:
  - Alert Messages are generated by an AV.
  - The Alert Message includes also the estimated tx range for that hop.
  - A node receiving the Alert Message waits for a time that is proportional to the node's position with respect to the estimated maximum tx range.



ROFF (Robust Fast Forwarding)

- Description: A multi-hop deterministic delay-based method to ensure robust communication.
- Mechanism:
    - Vehicles send periodic hello messages for neighbourhood discovery.
    - The gathered information aids in dynamic transmission rate adjustment, enhancing reliability.

Probabilistic methods:

- Pros: High reliability.
- Cons: Increased end-to-end delay due to message retransmission.

Deterministic methods:

- Pros: Predictable end-to-end delay.
- Cons: Lower reliability due to potential node failures.

In general, when dealing with VANETs choosing a deterministic approach will result in multiple collisions and possible problems for the network.

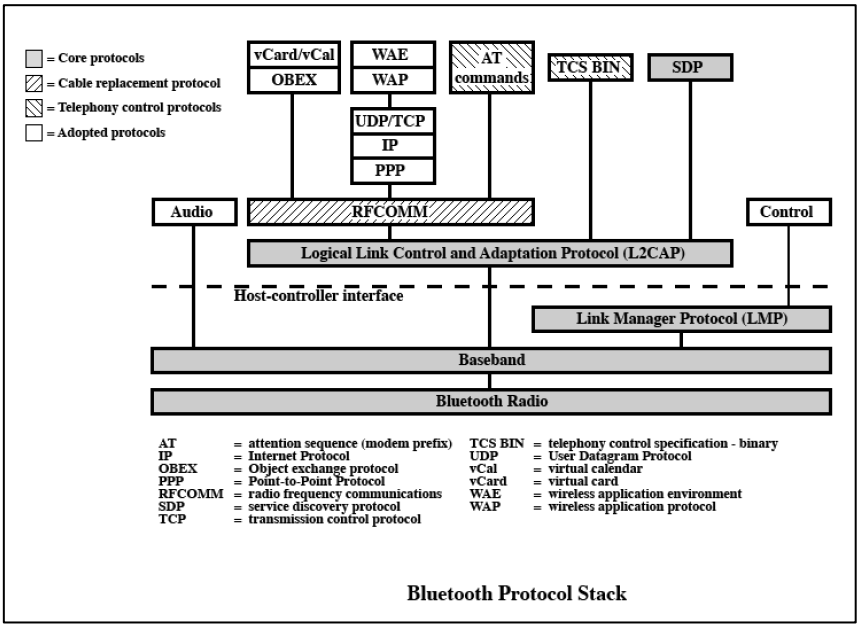
8. Bluetooth

Bluetooth is a standard initially created as cable replacement protocol over short distances. Several companies worked with Bluetooth over the time, companies such as Ericsson, IBM, Nokia, Toshiba, etc. Initially every company worked independently from the others until they decided to join forces and efforts to create a common standard, able to support compatibility among devices of different brands. Nowadays almost all connected devices have Bluetooth.

One of the Bluetooth use cases is to provide **synchronization among devices**: automatic synchronization of calendars, address books, business cards; push button synchronization (also known as “active synchronization”); proximity operation.

Applications

- Cordless Headset
- Portable PC speakers
- Cordless printer, scanner, keyboard, mouse, LAN
- Synchronization among devices
- Internet bridge
- Direct file sharing among devices
- Ad hoc networking (Bluetooth allows devices to connect wirelessly and extend their communication range by forming a temporary, self-configuring network without relying on a central infrastructure)
- IoT



Bluetooth Protocol Stack

Just to know, Bluetooth is not simple as it seems since it is characterized by several protocols that work together to provide Bluetooth functionalities. Bluetooth can be used to support different kinds of applications, in that case only a subset of the entire Bluetooth Protocol Stack is used to meet specific requirements.

Bluetooth is a **layered protocol architecture**, made up by core protocols, cable replacement and telephony control protocols and adopted protocols.

Core protocols

- Radio: details of air interface (frequency hopping, modulation, transmission power).
- Baseband: connection establishment in **Piconet** (the network created by Bluetooth nodes), packet format, timing, addressing.
- Link manager protocol (LMP): setup between Bluetooth devices, security aspects, authentication, encryption.
- Logical link control and adaptation protocol (L2CAP): adapts upper-layer protocols with Baseband.
- Service discovery protocol (SDP): device information, services. Used to establish a connection between two or more Bluetooth devices.

Cable replacement protocol

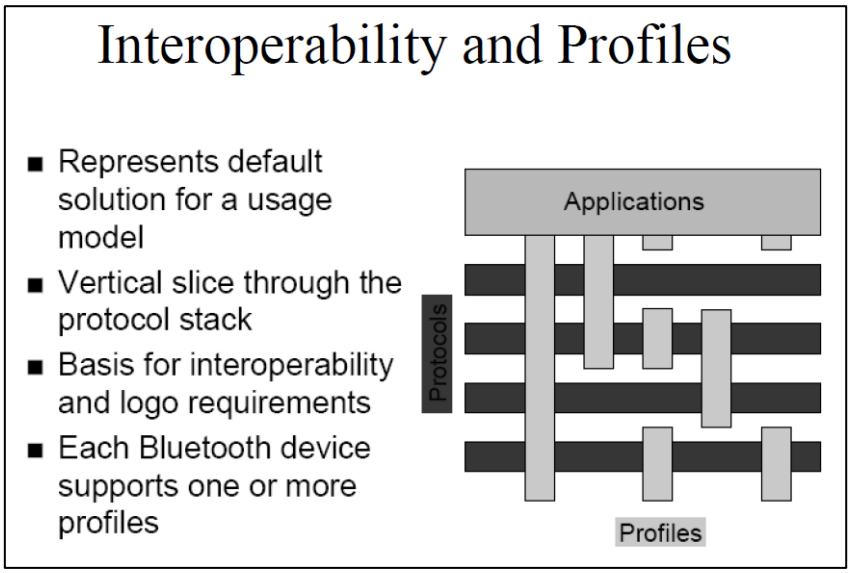
- RFCOMM (virtual serial port).

Telephony control protocol

- Telephony control specification –binary (TCS BIN).

Adopted protocols (use existing protocols and invent new ones only when necessary)

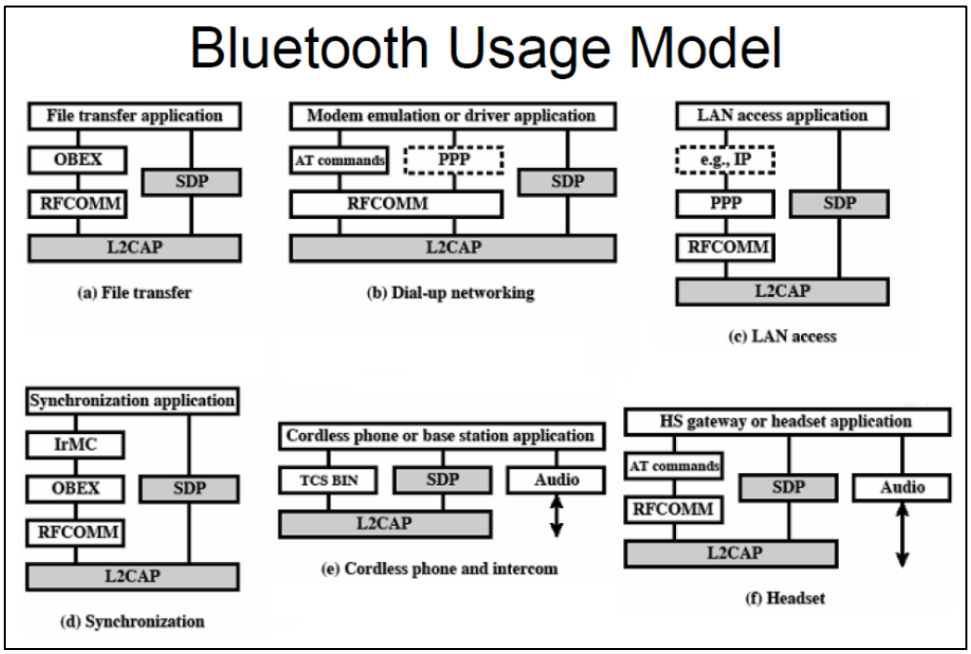
- PPP (Point-to-Point protocol, IP datagrams over a point-to-point link).
- TCP/UDP/IP.
- OBEX (Object Exchange protocol, defines objects and operations): defined by Infrared Data Association (IrDA).
- WAE/WAP (Wireless Application Environment/Protocol).



Bluetooth technology offers various usage models that enhance device connectivity and functionality. It is possible to count several usage models, such as:

- **File transfer** enables seamless sharing of documents, photos, and media between devices.
- **Internet bridge** model allows devices to use a Bluetooth connection to access the internet through another device, such as a smartphone.
- **LAN access** facilitates connections to local area networks, supporting efficient data sharing and remote access in local environments.
- **Synchronization** keeps data across devices up-to-date, ensuring consistency in calendars, contacts, and files.
- **Three-in-one phone** model merges the capabilities of a mobile, cordless, and intercom phone, providing flexibility in communication.
- **Headsets** allow users to engage in hands-free calls and media playback.

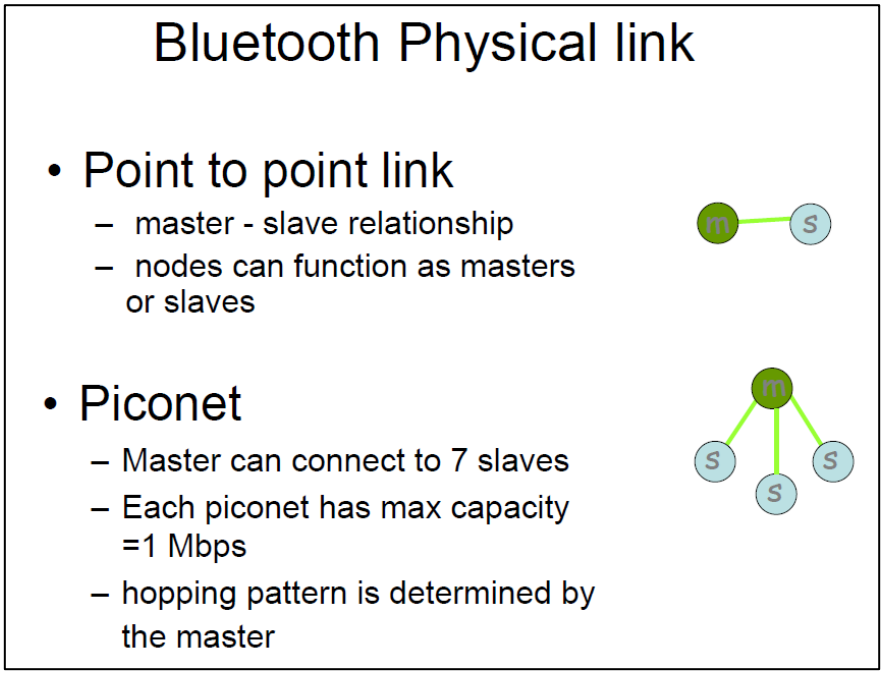
Here below we can see how different profile are used to implement the different usage models.

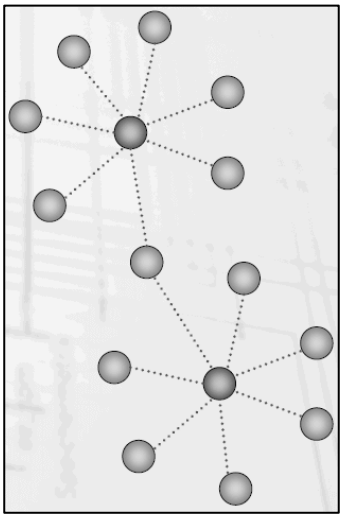


Bluetooth Radio and Baseband Parameters

- Topology: up to 7 simultaneous links in a logical star
- Peak data rate: 1 Mbps
- RF Band: 2.4 GHz
- Frequency allocations (in US and EU): 2.4 to 2.4835 GHz
- RF Channels: 2.402 GHz + n MHz (n = 0, ..., 78)
- Carrier Spacing: 1 MHz
- Transmit power: 0.1 mW
- Piconet access: FH-TDD-TDMA
- Scatternet access: FH-CDMA
- Hop Rate: 1600 hops per second (625µs - microseconds - duration per hop)

A **piconet** is a collection of devices connected in an ad-hoc fashion via Bluetooth. One unit/node will act as a **master** and the others as **slaves** for the duration of the piconet connection. Slaves can communicate between each other, but they need to pass through the master before. Master sets the clock and hopping pattern. Each piconet has a unique hopping pattern/ID pseudo-randomly chosen. Each master can connect to 7 simultaneous or 200+ inactive (parked) slaves per piconet.



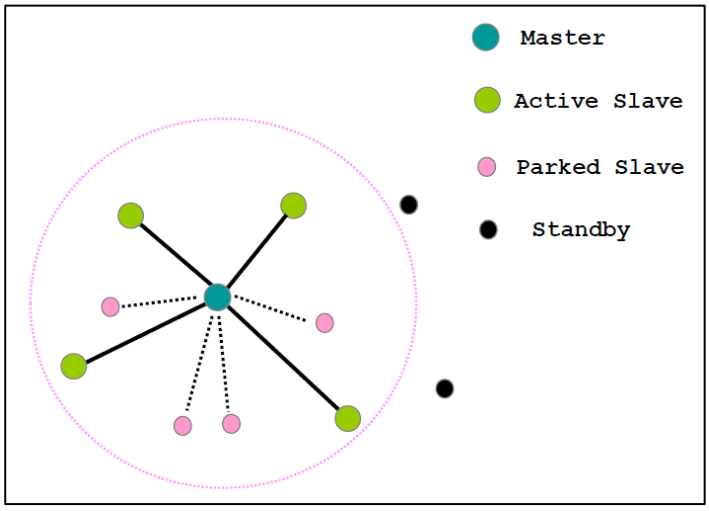


The network topology may be more complex, resulting in **multiple piconets interconnected each other**. There is one master per piconet. Some devices can participate in multiple piconets, allowing them to act as a bridge between separate piconets and expanding the network. This means that a node may belong to piconet i at a certain time t_i and to piconet j at time t_j.

These piconets operate in an ad-hoc manner, meaning they do not rely on a fixed central structure but rather form spontaneously as devices come into range.

A piconet is the basic unit of Bluetooth networking, composed by a master and one to seven slave devices. The master determines channel and phase.

A **scatternet** instead are those networks where more piconets are merged. Device in one piconet may exist as master or slave in another piconet. It allows many devices to share same area. Scatternet makes efficient use of bandwidth.

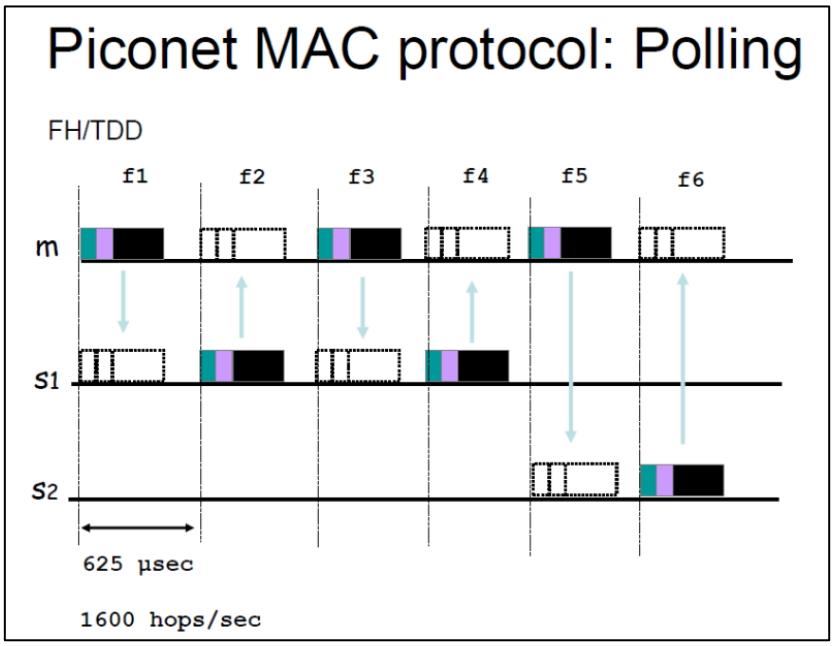


Inquiry scan protocol is used as first step of connection setup. It allows to hear other near devices, to learn about the clock offset and device address of other nodes in proximity. **Page scan protocol** is then used to establish links with nodes in proximity.

Parked slaves are those slaves not active in piconet. Over the time there might be some changes, and that node may pass to active mode. *Standby nodes* instead are out of piconet transmission range.

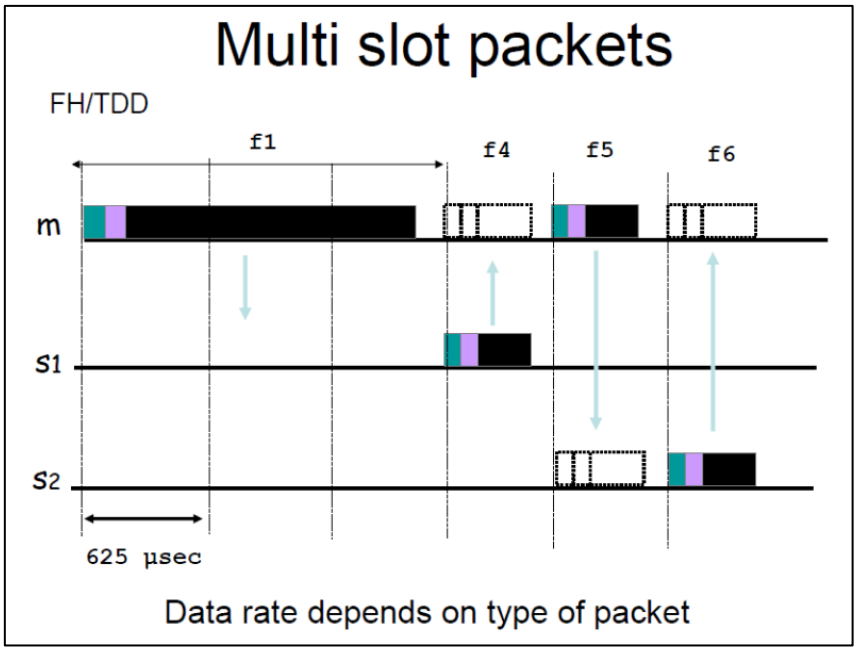
Addressing

Bluetooth addressing includes multiple types of addresses to manage device identification and connectivity within a network. Each Bluetooth device has a unique **Bluetooth Device Address (BD_ADDR)**, which is a 48-bit IEEE MAC address that serves as a permanent identifier for the device, similar to a traditional network MAC address. For active communication within a piconet, an **Active Member address (AM_ADDR)** is assigned to each connected device. This is a 3-bit address that identifies active slaves within the piconet, while an all-zero AM_ADDR is used for broadcasting to all devices. Devices in a low-power mode, known as "parked," are assigned a **Parked Member address (PM_ADDR)**, which is an 8-bit address allowing the device to remain connected to the piconet without actively participating in data transfer.
These distinct addressing mechanisms help Bluetooth manage both active and parked devices within its ad-hoc network structure efficiently.



The master continuously forwards messages to specific slaves, and it asks slaves whether they have something to transmit in a sort of round robin way. Since the master controls all the

piconet communications there is no collision (in the above figure note that the arrows are always between m and s_i). The master has complete freedom, it can decide what to do and how to do. For instance, master can also occupy more than a slot, resulting in higher data rate, depending on the packet type.



Error Correction Schemes

- 1/3 rate FEC* (Forward Error Correction): used on 18-bit packet header, voice field in HV1 (High quality Video) packet.
- 2/3 rate FEC: used in DM (Data Message) packets, data fields of DV (Data Video) packet, FHS packet and HV2 packet.
- ARQ: used with DM and DH packets.
    - Error detection: destination detects errors, discards packets.
    - Positive acknowledgment: destination returns positive acknowledgment.
    - Retransmission after timeout: source retransmits if packet unacknowledged.
    - Negative acknowledgment and retransmission: destination returns negative acknowledgement for packets with errors, source retransmits.

*FEC is a technology that allows the receiver to autocorrect certain errors of the received packets. Especially used in satellite communications where retransmitting the packet consumes a lot of time. In Bluetooth there are different types of FEC, depending on the reliability you want to achieve for each packet type (e.g., DM, DV, HV1, HV2, etc.).

Many Bluetooth versions

- Bluetooth 1.0
- Bluetooth 1.1
- Bluetooth 1.2
- Bluetooth 2.0 (up to 3 Mbps and reduced latency)

- Bluetooth 2.1
- Bluetooth 3.0 (improved speed and cooperation with Wi-Fi)
- Bluetooth 4.0
    - LE: Low Energy (speed up to 1Mbps)
    - UWB: Ultra Wideband
- Bluetooth 4.1 (improved interaction with 4G LTE)
- Bluetooth 4.2 (improved interaction with IoT)
- Bluetooth 5.0 (improved range, speed and interference avoidance)

8.1 IEEE 802.15.4 ZigBee

ZigBee technology is designed to meet market needs with several key advantages. It requires no new wires, simplifying deployment in existing infrastructures. **Installation and maintenance are easy**, with a mesh and self-organizing structure that reduces manual setup. The technology is **reliable**, supporting multiple channels and interference tolerance to ensure consistent performance. **Security** is provided through AES-128 encryption, protecting data transmissions. ZigBee is highly scalable, supporting hundreds of thousands of devices, making it suitable for large networks. It features **low power consumption**, enabling devices to "sleep" for extended periods and potentially last years on batteries. Additionally, ZigBee offers a **low-cost** solution with a small footprint, making it an affordable option for a range of applications.

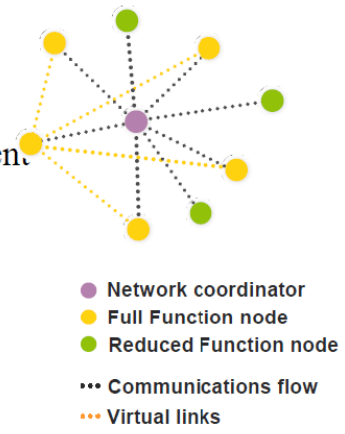
ZigBee operates as a high-level communications protocol, utilizing small, ultra-low-power digital radios based on the IEEE 802.15.4 wireless standard. Its targeted applications include secure networking, long battery life, and low data rate communications, making it ideal for applications requiring efficient, long-lasting connectivity. ZigBee operates on several RF bands: it uses 2.4 GHz with 16 channels for global applications, 915 MHz with 10 channels for North America and Australia, and 868 MHz with 1 channel for Europe. These options provide flexibility to support ZigBee across various regions and use cases.

ZigBee	Bluetooth	WiFi
802.15.4 standard	802.15.1 standard	802.11 standard
250 kbps	1 Mbps	< 54 Mbps
TX: 35 mA	TX: 40 mA	TX: 400+ mA
Standby: 3 uA	Standby: 200 uA	Standby: 20mA
32-60 KB memory	100+ KB memory	100+ KB memory
Lighting, sensors, RC peripherals	Telecom audio, cable replacement	Enterprise, home access points
Mesh networking	Pt-Mpt	Pt-Mpt

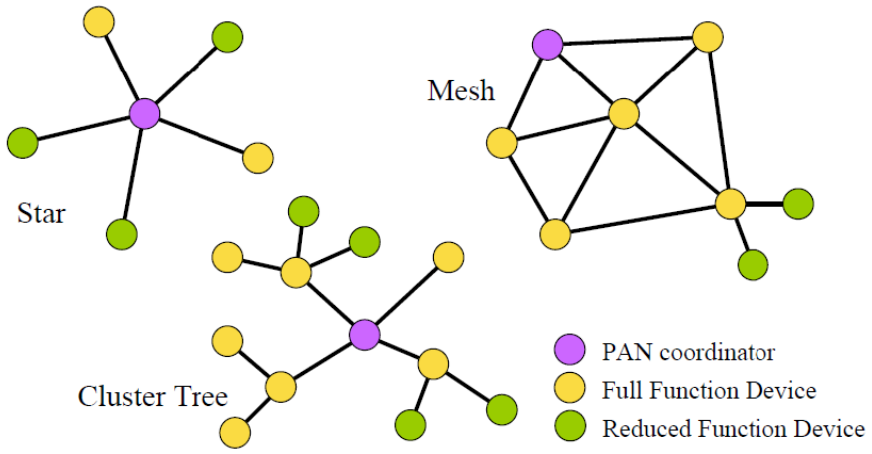
ZigBee finds most of its applications in sensors & controls in different scenarios, such as home automation, industrial automation, remote metering, automotive networks, interactive toys, medical.

Basic Network Characteristics

- 65,536 network (client) nodes
- Optimized for timing-critical applications and power management
  - Time to Join Network: <30ms
  - Sleeping to active: <15ms
  - Channel access time: <15ms
- Full Mesh Networking Support



ZigBee Network Topologies



ZigBee Device Types

- ZigBee Coordinator (ZC)
  - One and only one required for each ZB network.
  - Initiates network formation.
  - Acts as 802.15.4 2003 PAN coordinator (FFD).
  - May act as router once network is formed.
- ZigBee Router (ZR)
  - Optional network component.
  - May associate with ZC or with previously associated ZR.
  - Acts as 802.15.4 2003 coordinator (FFD).
  - Participates in multihop routing of messages.
- ZigBee End Device (ZED)
  - Optional network component.
  - Shall not allow association.
  - Shall not participate in routing.

Whether ZigBee and Bluetooth are competitors or complements?

Bluetooth seems best suited for:

- Synchronization of cell phone to PDA
- Hands-free audio
- PDA to printer

While ZigBee is better suited for:

- Controls
- Sensors
- Lots of devices
- Low duty cycle
- Small data packets
- Long battery life is critical

Timing Considerations

ZigBee:

- New slave enumeration = 30ms typically
- Sleeping slave changing to active = 15ms typically
- Active slave channel access time = 15ms typically

Bluetooth:

- New slave enumeration = >3s, typically 20s
- Sleeping slave changing to active = 3s typically
- Active slave channel access time = 2ms typically

Conclusion: ZigBee devices can quickly attach, exchange information, detach, and then go to deep sleep to achieve a very long battery life. Bluetooth devices require about ~100X the energy for this operation.

Power considerations

ZigBee:

- 2+ years from "normal" batteries.
- Designed to optimize slave power requirements.

Bluetooth:

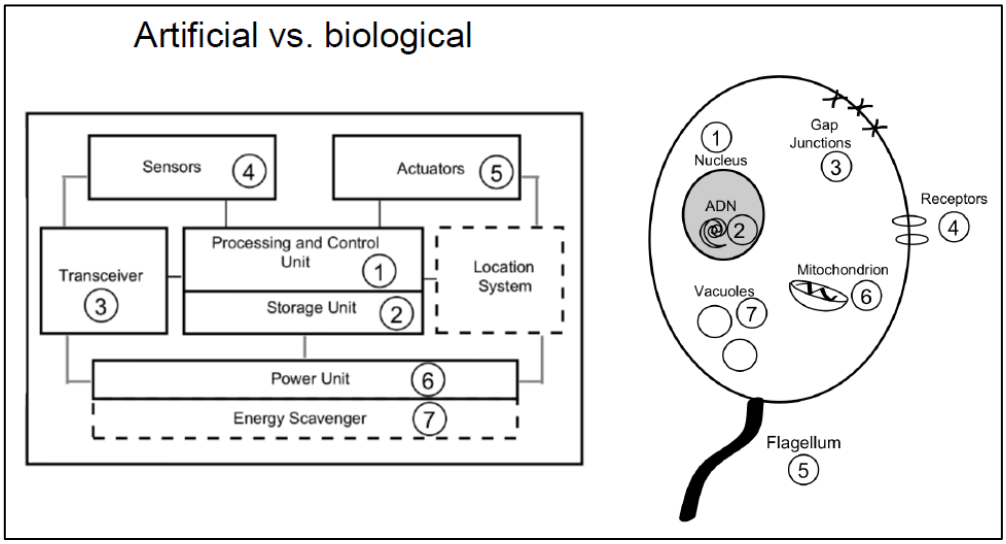
- Power model as a mobile phone (regular daily charging).
- Designed to maximize ad-hoc functionality.

The Bluetooth consortium has been working on a low-power version of Bluetooth called **Wibree** since 2001. Wibree offers performance similar to ZigBee, with Bluetooth wireless technology but without frequency hopping and the ability for nodes to remain asleep most of the time. Wibree has been adopted into the official Bluetooth specifications and will use the same hardware as standard Bluetooth devices, sharing the antenna.

	Bluetooth	Wibree	ZigBee
Band	2.4GHz	2.4GHz	2.4GHz, 868MHz, 915MHz
Antenna/HW	Shared		Independent
Power	100 mW	~10 mW	30 mW
Target Battery Life	Days - months	1-2 years	6 months - 2 years
Range	10-30 m	10 m	10-75 m
Data Rate	1-3 Mbps	1 Mbps	25-250 Kbps
Component Cost	$3	Bluetooth + 20¢	$2
Network Topologies	Ad hoc, point to point, star	Ad hoc, point to point, star	Mesh, ad hoc, star
Security	128-bit encryption	128-bit encryption	128-bit encryption
Time to Wake and Transmit	3s	TBA	15ms

9. Molecular communication

Molecular communication is a type of communication that happens between nano-scale objects/devices, shaping so the so called "**nano networks**". A **nano-scale device** (or machine) is a simple device able to perform a specific task at nano-level. Tasks are very simple and restricted, such as communicating, computing or data storing. Nano-scale devices may be artificial and naturally.



There are different approaches to build nano-machines.

- **Top-down approach**: development of nano-scale objects by downscaling current existing micro-scale level components.
- **Bottom-up approach**: nano-machines developed using individual molecules. Anyway, manufacturing technologies able to do that do not exist, yet. Instead of starting with something big and making it smaller, this approach involves building nano-machines from scratch, piece by piece, starting with individual molecules.
- **Bio-hybrid approach**: biological nano-machines as models or building blocks to develop new nano-machines. The bio-hybrid approach uses these natural nano-machines as inspiration or even as components. Scientists either copy how these biological systems work or directly integrate them into man-made nano-machines.

A **Nano-network** is just a set of components on a nano-scale and their interconnection. Probably more biological than electronics, this because nowadays it is hard to build something at a nano scale and we also need to consider some physical properties. In fact, we can identify three macro-categories of communication media: standard communication, nano-mechanical communication, molecular communication.

STANDARD COMMUNICATION

Electromagnetic waves are a common method of communication, but they face challenges at the nano-scale. Wiring large numbers of nano-machines is impractical due to their tiny size, so wireless solutions seem like a better fit. However, integrating antennas into nano-machines is extremely difficult, as they are often too large relative to the machines. Moreover, the energy required to power these antennas is far beyond what nano-machines can typically provide.

Another possible communication method is through **acoustic waves**, where nano-machines would use tiny transducers to detect and respond to sound waves. However, the primary challenge lies in creating transducers small enough to fit into nano-machines. This size constraint makes implementing acoustic communication highly complex and not yet feasible.

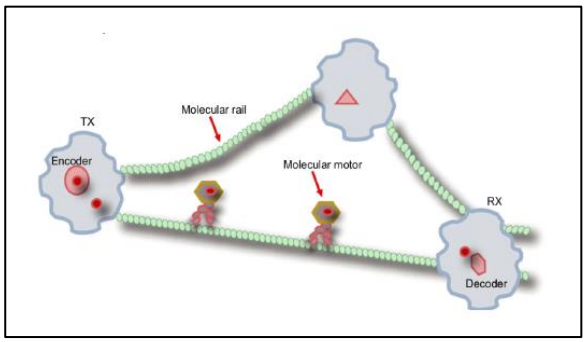
NANO-MECHANICAL COMMUNICATION

Information is transmitted through hard junctions between linked devices at nano-level. The main drawback is that this solution requires physical contact between transmitter and receiver.

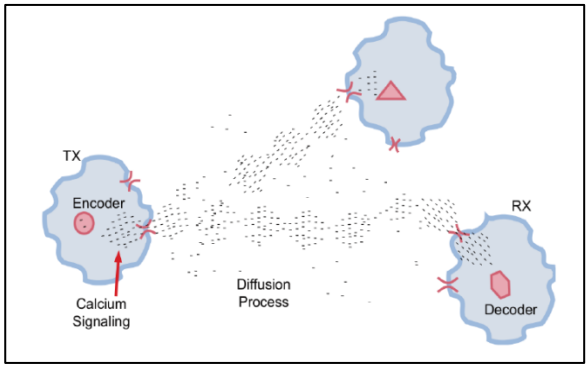
MOLECULAR COMMUNICATION

Molecular communication involves transmitting and receiving information through molecules, making it one of the most promising approaches for nano-networking. Unlike other methods, molecular transceivers have already been conceptualized at the nano-scale, enabling communication between transmitters and receivers located as far apart as the molecules can travel.

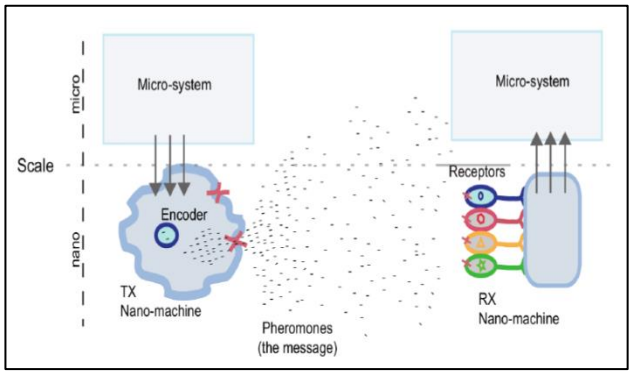
For short-range communication, within millimetres, mechanisms like molecular motors and calcium signalling are used. For longer distances, ranging from millimetres to kilometres, systems like pheromones or bacteria can carry molecular messages effectively.



Molecular Motors are proteins that transform chemical energy into mechanical work. They travel along molecular rails called micro-tubules, previously deployed setting a complete railway. Just like as standard communications, data need to be encoded, transmitted and decoded.



Calcium Signalling is one of the most well-known molecular communication technique in biology. Responsible for many coordinated cellular tasks such as fertilization, contraction and secretion More flexible, there is no need of railways as in molecular motors communication. Similar to broadcast networks, all surrounding nano-machines can receive a broadcast message.



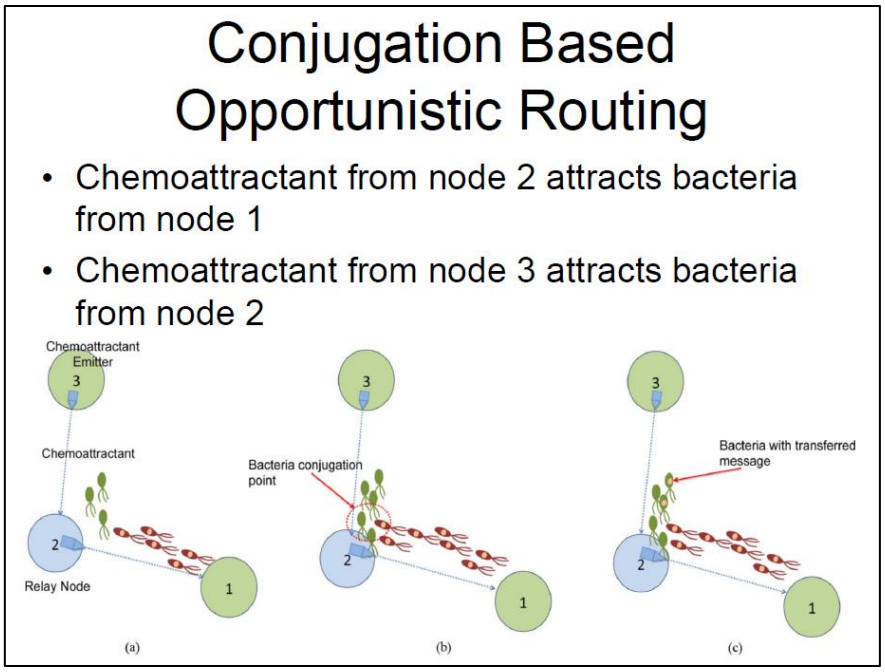
Pheromones are molecular compounds that carry information, which can only be interpreted by specific receivers. These messages consist of molecules, allowing for an enormous variety of possible combinations. In nature, such as in ant colonies, entire communication systems rely on pheromones to coordinate activities and maintain social organization.

Bacteria have a number of interesting characteristics, such as conjugation, chemotaxis, antibiotics resistance. Bacteria is the ideal as medium range information carriers for nano-networks.

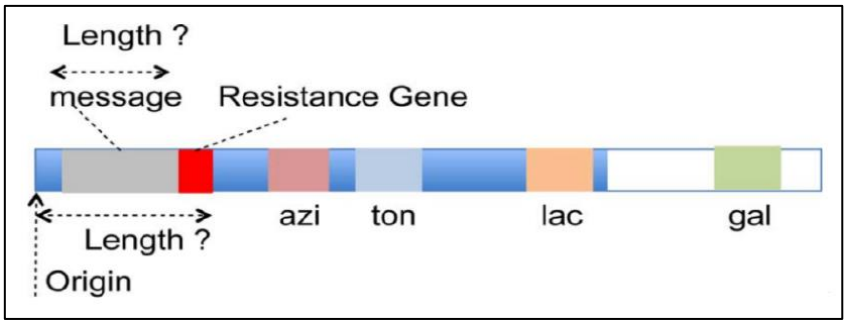
Bacteria conjugation allows different bacteria to interconnect and pass copies of plasmids. Plasmids are genetic messages encoded as strings of four nucleotides.

Bacterial chemotaxis is the process by which bacteria move in response to chemical stimuli. They can swim toward food sources, such as glucose, or move away from harmful substances, like salts. Bacteria also communicate with each other by emitting chemical signals, known as chemoattractant or chemorepellents. This form of communication allows bacteria to coordinate their movements and behaviours, which enhances the survival rates of their colonies by promoting cooperation among individuals in response to environmental conditions.

Conjugation-based opportunistic routing is a communication method suited for opportunistic networks, where network connections are intermittent or have highly variable performance. This approach is particularly applicable to bacteria-based nanonetworks, where, similar to wireless broadcast networks, messages that are not meant for specific recipients are simply discarded. In this system, bacteria use chemoattractant to guide the movement along a desired path. By combining conjugation, which involves the transfer of genetic material between bacteria, with chemotaxis, which directs bacterial movement in response to chemical signals, these networks can effectively route messages and coordinate actions within the colony.



Resistance to antibiotics is a key feature used in managing bacterial populations, particularly in the context of plasmids. Antibiotics can be employed as a selective tool to target and eliminate bacteria that lack the proper or complete plasmids needed for specific functions. Bacteria that have only undergone partial conjugation, and therefore do not possess fully functional plasmids, are effectively discarded by the use of antibiotics. This strategy ensures that only bacteria with the correct genetic material survive, maintaining the desired traits in the bacterial population.



Some possible applications:

- Novel healthcare and medical technologies
  - Targeted Drug Delivery [Okonkwo et al., 2016]
  - Live health monitoring
- Novel environment technologies
  - Oil spilling containment [De Lorenzo, 2001]
  - Water resources monitoring
- Counter bioterrorism applications
- Being highly critical applications, the implementations should be highly reliable and secure

10. Indoor localization

There a lot of services related to the device location, think about all you can do with a smartphone knowing your location (personalized services, etc.). There are different means that manage devices localization outdoor (e.g., GPS). Anyway, indoor localization is more challenging, we cannot use the same outdoor means, such as the GPS, so other tools are needed. Indoor localization is not yet completely solved, we will examine the main challenges and the proposed solutions.

To assess a technology for indoor localization we can use several metrics:

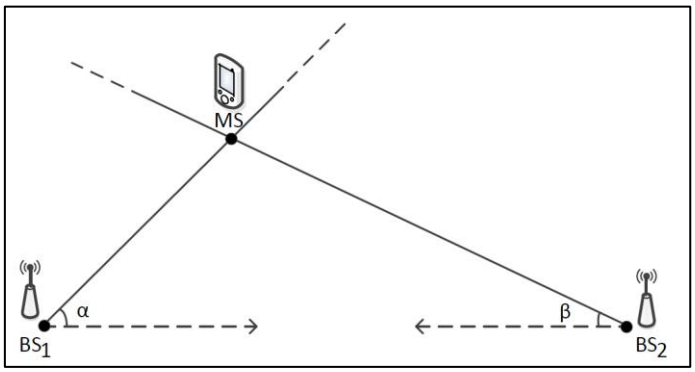
- **Accuracy**: average error between estimated measure and the actual one.
- **Precision**: error distribution regarding the actual position vs the estimated one.
- **Robustness**: the ability in maintaining accurate estimation even when changing the context/environment.
- **Scalability**: the system behaviour when changing the number and density of devices.
- **Cost**: includes the hardware, the initial set up, the maintenance.

Notation:

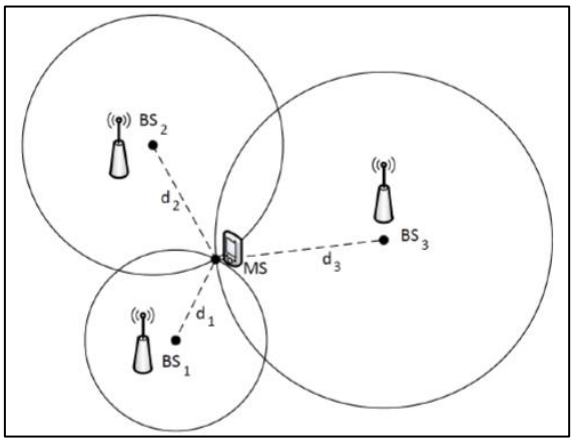
- We can assume to use a cartesian coordinate system in the monitored environment.
- Mobile Station (MS): is the device that needs to be localized (e.g., a smartphone).
- Base Station (BS): is an infrastructure component of the coordinate system (e.g., an Access Point).

The **Triangulation method** (or "Angle of Arrival") requires knowledge of the arrival angles of the signal emitted by the MS and received by the BS.

- At least two angles are needed to compute MS position.
- Requires complex hardware on the BS.
- Not really usable indoor since strong multipath effects in indoor environments that may lead to interference and wrong receiving angles measurement.



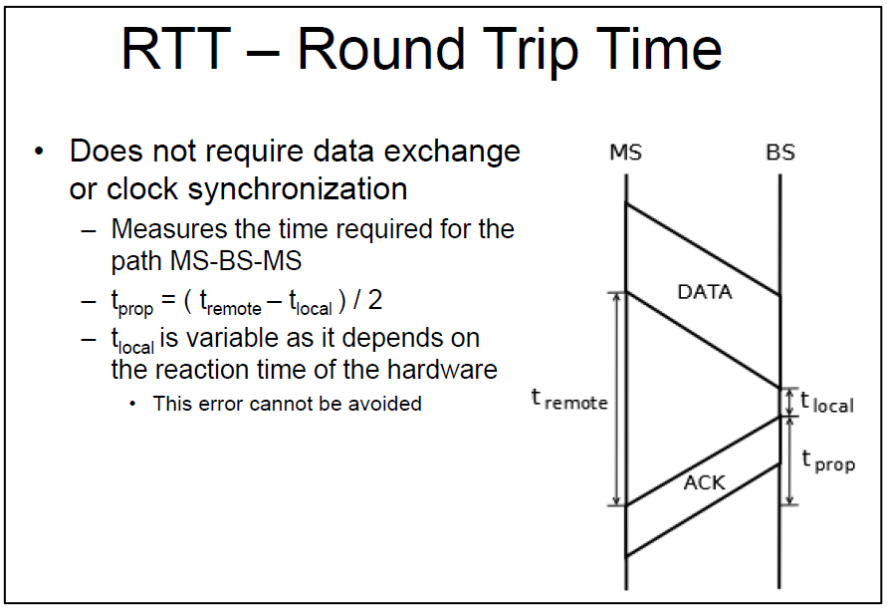
Trilateration requires knowledge of the distance between the MS and the BS. Required the distance with 3 or 4 BS for localization in 2D or in 3D, respectively.



The distance between MS and BS can be **estimated through the propagation time** of the radio signal. Since they are electromagnetic waves the propagation speed is **c = 3 × 108m/s**. Considering **t_prop** the measured propagation time, then the distance is **d = c × t_prop**.

Time of Arrival (ToA), our **t_prop**, requires MS and BS to have a synchronized clock (otherwise the time stamps for when the signal was sent and received will not align) and the possibility to exchange data.

1. BS emits the signal and sends to the MS the time **t1** at which the transmission ended.
2. The MS completes the reception of the signal at time **t2**.
3. The MS computes the propagation time as **t_prop = t2 – t1**.



The RTT calculation includes the hardware's reaction time (t_{local}), which can vary and introduce unavoidable errors. RTT assumes the signal takes the same amount of time to travel both ways (MS-BS and BS-MS). Any asymmetry in the path, such as delays caused by different hardware processing times or environmental interference, can reduce accuracy.

ToA remains the better choice for high-precision localization, as it avoids the inherent errors introduced by hardware reaction time and asymmetrical paths.

We need to consider the **measurement error** generated by the granularity of the timer used to measure the time. Most WLAN 802.11 boards allow to save the hardware timestamp of MAC layer packets with a precision of 1 µs, corresponding to a granularity of 300 m when considering the speed of light. This precision is not sufficient and two possible solutions are proposed:

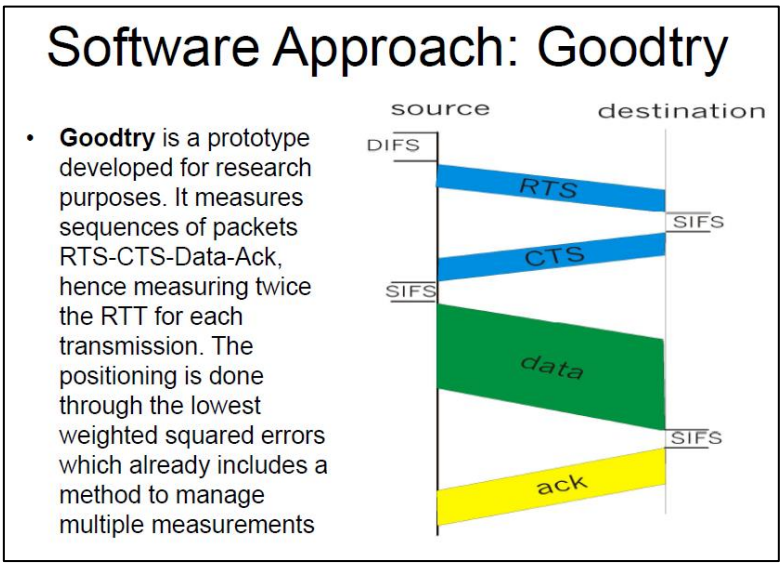
- **HW approach**: use the time stamp provided by enhanced/modified HW.
- **SW approach**: use multiple measurements to obtain an estimation close to the actual value.

Hardware Approach

Specific hardware able to use bits transmitted/received to trigger a MAC layer counter based on the clock of the WLAN board. This clock has a frequency of 44 MHz, which corresponds to a precision of 2.27×10^{-8} s → 6.82 m. Improved precision via statistical methods and multiple measurements.
Measurements are done at the lowest possible layer (MAC layer) to avoid variable delays in execution time induced by upper layers.
With 802.11 the RTT is measured with the Data-Ack pair of packets as the time interval between the reception of the former and the transmission of the latter is reasonably constant (SIFS).

Software Approach

In a software-based localization approach, hardware time-stamps are typically provided only for received packets by regular WLAN boards, not for sent ones. To overcome this limitation, a monitoring station is required, positioned as close as possible to the MS that we want to localize. This station listens to the communications between the MS and the BS to obtain consistent hardware time-stamps for both sent and received packets. However, the need for a dedicated monitoring station makes this approach impractical for real-world applications outside of academic settings, as it requires additional infrastructure and coordination, which can be challenging to implement in large-scale or dynamic environments.



TDOA (Time Difference of Arrival) is a technique that measures the arrival time of a signal emitted by the MS towards multiple BS. By exploiting the differences in arrival times at each BS, the position of the MS can be calculated. This method requires synchronization among the base stations to ensure accurate time measurements. For 3D localization, at least four BS are needed, while for 2D localization, three BS are sufficient. Additionally, a location server is required to manage both the synchronization process and the collection of measurements from the various BS. However, TDOA is not suitable for self-positioning, as the measurements can only be made at the BSs, making it dependent on external infrastructure for localization.

Scene analysis is a method that consists of two main phases:

1. Collection (and storage in a database) of fingerprints at predetermined and known locations. A fingerprint typically includes the measurement of the signal strength received by a MS from different BS (RSSI).
2. Collection of the fingerprint at the current, unknown position and comparison with the stored data in the database using AI algorithms (e.g., k-NN, SVM) or statistical methods.

Even with this approach, at least four BS are required for 3D localization, and at least three BS are needed for 2D localization.

Scene analysis methods require significant initial effort to create the fingerprint database. The **RSS (Received Signal Strength)** in indoor environments is influenced by:

- **Multipath:** due to reflected signals, the measured strength is higher than ideal.
- **Shadowing:** in the case of non-line-of-sight (NLOS), the signal strength is not easily computable as the transmitted frequency is also absorbed by water (and thus by people).
- **Moving objects:** these cause sudden fluctuations in the RSS, necessitating multiple measurements.

In the case of non-temporary variations in the environment, such as furniture rearrangements or changes in the position of the BSs, the offline training phase needs to be repeated.

k-Nearest Neighbour (kNN)

Given:

- m: the number of base stations (BSs).
- n: the number of fingerprints in the training set.
- S_i: the fingerprint (an array of size m) corresponding to the point (x_i, y_i, z_i) in the training set.
- s: the RSS measurement taken by the mobile station (MS) during the online phase.

The kNN algorithm calculates the distances between the current measurement s and every fingerprint in the training set. It then selects the k points from the training set with the smallest d_i^2 values.

The coordinates of the MS are estimated as the arithmetic mean of the coordinates of the selected k locations, or the weighted mean, based on the distances of the selected k points.

RFID (Radio Frequency Identification) systems consist of the following components:

- RFID Reader: A device that emits a signal to query nearby tags and receives their IDs in response.
- RFID Tag: A device that responds to the reader's query by transmitting its unique ID. Tags can be:
    - Passive Tag: Low cost (~$0.30), shorter range, longer lifespan (no battery).
    - Active Tag: Higher cost (~$3.00), longer range, shorter lifespan (battery-powered).

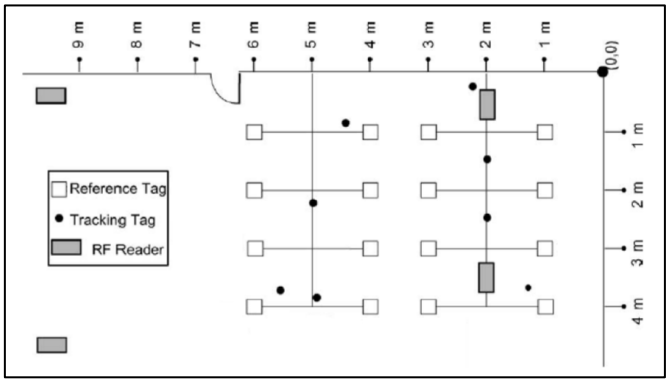
The most expensive component of an RFID system is typically the reader, which costs around $1,000.

LANDMARC is a positioning system that uses active RFID, and a scene analysis method based on RSSI. It consists of:

- **RFID Readers:** functioning as base stations (BSs) with the ability to communicate measured data to a localization server.
- **Reference Tags:** RFID tags with known coordinates.
- **Tracking Tag:** the RFID tag to be localized (acts as the mobile station, MS).

In this system, a fingerprint is defined as the array of RSSI values of signals emitted by RFID tags and received by the BSs.

Unlike traditional systems, LANDMARC does not require an offline training phase, as the fingerprints of the reference tags are dynamically measured. This approach enhances the system's robustness to changes in the environment.



Localization in LANDMARC using kNN

The kNN algorithm is used for localization, leveraging the fingerprints of reference tags as the training set.

- The system's performance is heavily influenced by the hardware, as not all RFID readers provide sufficiently fine granularity for RSS measurements.
- Active RFID tags are battery-powered, and the system requires that the transmission power of all tags (reference and tracking) be very similar.
- To ensure this, RFID tags of the same type and with the same battery level must be used, allowing for comparable measurements suitable for localization.

Passive RFID for Localization

Using an RFID reader as the mobile station is a less expensive solution compared to active RFID when dealing with wide spaces and a need to localize only a few nodes. Employing passive RFID tags as reference tags further reduces costs. This approach relies on fingerprints, but the training phase can be resource-intensive, requiring approximately 2,000 snapshots for a 50 m^2 area. Each snapshot is derived from multiple measurements at the same location to ensure accuracy. However, this system is not very robust against environmental changes, making it most suitable for controlled settings like robotics or automated environments.

AR may also be used for device localization. Augmented Reality solutions for localization use a combination of real-time data from cameras, sensors, and environmental markers to determine a user's position and overlay digital information onto the physical world. By integrating visual features from the environment and combining them with GPS or indoor positioning systems, AR can provide accurate localization in both outdoor and indoor settings, enabling applications in navigation, gaming, maintenance, and training.