WIRELESS SYSTEM

Wireless systems have been designed to enable communications anywhere and anytime.

CONS:

- **Mobility** must be supported as users move during conversation and we need to known their position,
- Portability must be supported since devices rely on batteries so we should optimize as much as possible the **lower power consumption,**
- The radium spectrum is divided into portion of frequency so in general you can transmit with a **limited capacity** (limit bandwidth)
- You have a **high Bit Error Rate (BER)** (in Wireless if you lose information is due to bad channel, bits flipping or you are not able to correct errors) so you need to be able to identify if you correctly receive the packet (**error detection capability**) and in case to re-transmit it. To avoid it you can consider:
  - Adding extra redundancy bit applying FEC while transmitting,
  - Use the Interleaving in which you divide a sequence of bits into several units and divide each unit in portion and then send packets obtained by putting together different portions of units. At the destination you reconstruct the original message (it reduce bit flipping and allow FEC to be more effective, but add latency)
- Wireless system is Broadcast so we need to take into accounts MAC layer to arbitrate the access to the medium to avoid **collisions** and I must handle the access from users to a share medium (**Shared channels**) and **Security** problem since everybody can listen while I'm transmitting.
- Devices are not full-duplex so they cannot transmit and receive simultaneously, so not use CSMA/CD with collision detection, but CSMA/CA in which you use an explicit ACK that the destination has received the message

PRO:

- Do not need cost of cabling a wired infrastructure
- You can move remaining connected and access to information anywhere and anytime

Infrastructure topology:

- In which devices transmit information to smarter device (Access Point or BSS) which is connected via Internet or wired network,
- Devices exchange information wirelessly among each other (ad hoc wireless Network) e.g., devices in disaster recovery phase
- Solutions in the middle

Moreover, if you consider ad hoc or mesh network topology the transmission pass through intermediate devices that act as touter, but in IOT case these devices alternate active/Inactive times so the **topology is dynamic** and I need to be able to route information to its destination.

EDGE COMPUTING: in case of IOT devices it allows energy consumption to be reduced by processing data on site and sending less data. In general IOT devices have limited capabilities and cannot process locally data since they should be low power designed, simple and inexpensive.

Wireless Transmission

To transmit over wireless channel is used an Antenna (Tx or Rx) which modulate the information (in bits) into electromagnetic waves so that you can transmit more information in a unit of time (higher data rate) but with more bit flipping problem.

When a signal propagates over the channel is subject to:

- Attenuation as function of the distance Tx-Rx
- Attenuation due to obstacles and so the signal can go through multiple paths (multipath fading problem).
  IDEAL CASE: Line of Sight (LOS) no obstacles
  REAL CASE: Reflection, Shadowing, Diffraction, Scattering → **Multiple path fading**

In LOS situation you can describe the **received power** as a curve which decay exponentially with the distance. In Real case you can have some variabilities as noises in the curve due to attenuations and obstacles so if you zoom in the curve you can see some oscillations ("fading" effects).

Considering the type of **Antenna**, in the Ideal case we consider the antenna as a isotropic radiator which emits the transmit power in all possible direction uniformly, in Real case it is not but it can approximate it: you can consider a omnidirectional dipole antenna (widely used) or a directional antenna which instead radiates in a given direction, with much power (but more complex). → **antenna gain** characterize the directivity of the antenna.

Then, you can calculate the received power (**Friis equation**) under LOS condition accounting for the Tx power, the distance d, the attenuations (geometrical spreading) and also thee frequency used for the transmission, the gains of the antennas and additional losses (HW losses) as:

$$P_{Rx} = P_{Tx}\, g_{Tx}\, g_{Rx} \left(\frac{wavelenght\ \lambda}{4\pi d^2}\right) \frac{1}{additional\ Losses}$$

You can generalize it in a Real Scenario considering a propagation coefficient η which depends on the propagation environment (e.g. =4 urban):

$$P_{Rx} = P_{Tx}\, g_{Tx}\, g_{Rx} \left(\frac{\lambda}{4\pi}\right) \frac{1}{d^{\eta}}$$

To represent how the curve decay over distance you can also consider the **Path Loss** = the ratio between the power of the Tx and the power at the Rx (in dB).

If received power is below a given threshold, information cannot be correctly received, for this reason you consider the **signal-to-noise ratio (SNR)** at a given distance. When it decreases (signal attenuation) the **BER** increases.

In general the quality of the channel can vary over time and you can use modulation to improve the quality of the transmission. The performance depends on: BER, channel conditions, SNR and on the physical layer I'm using.

Multiple path fading: while propagating the signal can follow multiple paths. Signal can pass through different effects (reflection, diffraction, …), the different replicas of the signal will be received at different times and they will be more or less attenuated. The results of what I receive depends on: the number of replicas, their phases, amplitudes and frequency.
There are models that represent this multipaths fading. One is the statistical model **Rayleigh fading**: It assume that the magnitude of a signal will vary randomly, or fade, according to a Rayleigh distribution.

Multipath fading introduced another problem: "**delay spread**". This means that the replicas are received spread a bit in time, so I have the shape of the signal changed, translated, attenuated and so on, and the duration is also longer and can overlap with the bandwidth of the following sign leading to **inter-symbol interference** (ISI). The impact of the Delay spread can be quantified by computing the root mean square.

Techniques for ENERGY efficient communications

When we talk about energy efficiency I consider the **network lifetime** (e.g. due to battery or disconnections) and the **energy consumption** during communication to transmit/receive packets, receive data and control packets, or when it is idle, ready to receive; and computational cost related in how the system operate, so we need to design a low power protocol to get a well working network.
In terms of networking stack there are different techniques adopted at each single layer then have to be combined, so at the end you consider a global optimization → **cross layer design solutions**.

- Avoid energy waste in protocol design: minimize collisions, avoid Transmission in bad channel conditions, header compression, data aggregation to reduce in network data transmission (Layers)
- Minimize the overall energy consumption during routing, also accounting for retransmissions; avoid passing through critical nodes;
- Energy aware routing solutions which account for residual energy (and expected future availability of energy in case harvesting is an option) when selecting the best next hop relay.
- **Hardware** selection can also have an impact on overall system power consumption.

PHY layer:

- I can use an **Adaptive modulation scheme**: active the Tx using a high data rate modulation to transmit data and afterward switch it to a low power mode "sleep mode"when transmission of non-addressed packets is detected.
- We can decrease the overall energy consumption in the case of long-range communication by applying **power control.**

MAC layer:

awake/asleep schedule: in which nodes consume energy when awake and can transmit data, while consuming less energy when asleep, but packets cannot be received or transmitted. The way to express this is computing the $duty\ cycle = \frac{Time\ ON}{Time\ ON + Time\ OFF}$
To avoid energy waste:

- Nodes not involved in communication should go to sleep till current information exchange completes,
- Nodes should **minimize collisions**, because otherwise I have to retransmit,
- **Header compression:** reducing the Header expressing the same info in a more efficient way, transmitting less bits (so Tx is ON for less time)
- Limit control information exchanged, aggregate redundant information

There are 2 **MAC protocols** designed to account to the fact that nodes are not always ON due to duty cycle:

- **Synchronous:** assume all devices to synchronize awake/asleep schedule, so I know when the device will be ON, so just waiting for this to transmit the information;
- **Asynchronous:** awake/asleep schedule is unknown, a sequence of packets must be sent continuously until the destination node wakes up and answers

Problem: the device is spending much time in the so called ***idol listening***: having to periodically being ON just in case someone has to transmit information. A possible way to reduce energy in that case, without having this limit associated with the duty cycle, is by adding a specific HW component:

Wake-up Radio, who has a very low power consumption, and It stays ON all the time and if you transmit a specific signal (*wakeup signal*) it is possible to recognize this.

Data Link layer:

It make the communication at LLC reliable, checking for errors during transmission, using FEC code and using automatic re-transmission scheme.

- To avoid waste, if the channel is in bad state is it convenient to delay re-transmission
- **ARQ** (Automatic repeat request) and **FEC** schemes have been studied to optimize energy consumption while ensuring reliable and timely communication.

ROUTING:

Depending on the scenario, it may be more energy efficient to transmit over a larger number of shorter links or to minimize the number of hops. (in short-range, more or less the consumption is the same). In general:

- Minimize the overhead associated to route discovery and maintenance
- Load balancing of the energy consumption among nodes to increase the network lifetime
- Energy aware routing solutions which account for residual energy (and expected future availability of energy in case harvesting is an option) when selecting the best next hop relay.
- Link quality aware relay selection to avoid retransmissions.
- Relay selection which prefers data fusion/aggregation.

Some studies compare energy consumption in 2G, 3G and Wi-Fi systems → **TAIL and RAMP energy effect:**

It accounts the energy consumed during time to switch ON to transmit (RAMP) and the energy consumed after the transmission ended but in which the device still remains in a high energy consuming mode (TAIL).

- To reduce the TAIL → **TailEnder**: transmit putting together a batch of communication, trying to reduce the percentage impact of the tail energy.

Results:

- the **GSM** (2G) devices consume less of **3G** devices, but in 3G there is the possibility to transmit at a higher data rate (so after a certain size of data, it is more efficient to use 3G). The **Wi-Fi** energy consumption is lower than the 2G and 3G since it is a short-rage, if we consider the control packets etc it increases but always below 3G.
- when the traffic increases, also the energy consumption increases. If we increase the data rate, we can be more effective and decrease the energy consumption.
- Important component comes-out by the experiment: **cross factor**, that depends on the frame rate (fps): basically, even if I split the data differently into more packets, and transmit using multiple single hop, I consume more.
  → Power consumption depends also on the cross-factor $P_{xg}(\lambda_g)$ → the idea is that by transmitting a single larger packet across the stack is more effective!

CELLULAR SYSTEMS

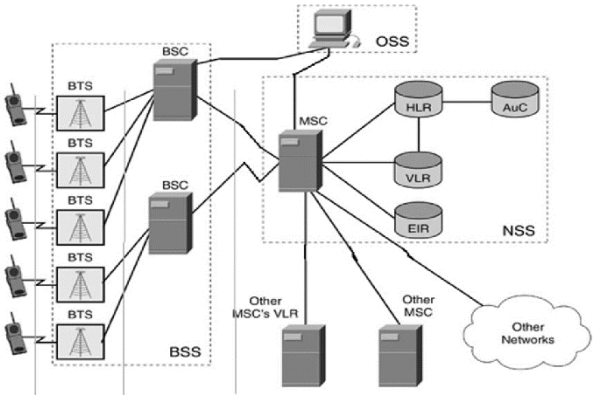
To perform a good sharing of the channels without collisions there are different solutions:

- **FDMA** (*Frequency*)**:** the available bandwidth is split in n channels, each one used by a single user.
- **TDMA** (*Time*): each user can use the entire bandwidth for a certain time slot.
- **CDMA** (*Code*): allows to share a band of frequency by using a chipping code in an orthogonal way to avoid collisions.
- **OFDMA** (*Orthogonal Frequency*): at each user is allocated a certain number of resource blocks in the time-frequency grid, allowing advanced dynamic resource allocation.

Frequencies Reuse: in general we have limited bandwidth and we need to optimize the users allocation. In a huge area we will have multiple users sharing the same resources and each of them could be able to exchange data with the cellular system operator.
It is important to REUSE these limited resource: at the beginning it has been done by clustering (frequency reuse) by splitting channels in sets, each with a different frequency so that users with the same frequency are far enough. The second generation (**GSM**) use a combination of **FDMA** and **TDMA** with a **full-duplex** approach: bidirectional communication in uplink (MS->BS) and downlink (BS->MS).

- Since we have mobile devices we want to minimize the energy consumption and maximize the lifetime, so we need **power control**,
- during pauses in speech transmission is interrupted to reduce interference and energy consumption.
- **Adaptive Equalization** is used to fix the problem of inter-symbol interference

GSM architecture



- **Mobile Station** (MS)
- **Base Station Subsystem** (BSS)
- **Network Subsystem** (NSS)

Types of communications:

- **User equipment (UE)**: Interfaces the user
- **Access network (AN):** Communication to/from the user equipment
- **Core network (CN):** communication between AN and external networks

The area where the GSM service is provided is called the **Public Land Mobile Network** (**PLMN**) **area**.
This is split in different subareas each managed by a single **MSC** and associated to a VLR temporary database. This area is split into subareas called **Location areas** (LA), identified by a LAI and in which we maintain location info about the users. LA includes several cells, and a **Cell** is the area covered by a single **BTS**, identified by a **BSIC** (Base Station Identity Code).

Mobile Station (MS): terminal owned by the user (e.g. smartphone). It is logically made of:

- ME (Mobile Equipment) through which we access the cellular network, identified by the IMEI
- SIM to identify and authenticate to the network and access to services. In SIM we have some information: Serial Number, IMSI, security authentication and ciphering information, Temporary Network information (LAI), PIN, PUK etc..

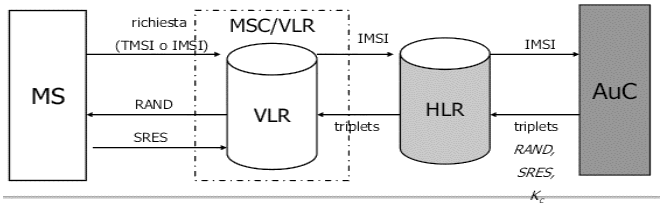
Base Station System (BSS): it allows communication to/from user over the radio link and with the MS to allocate physical resources of channels. BSS includes different logical elements:

- **Base Transceiver Station (BTS):** enable the transmission and reception of information through the radio interface to/from the MS. It performs:
  - signal modulation
  - encryption
  - quality measurements on power, BER, link quality.
  - It broadcast on a control channel the data needed for the MS to access the network (Cell identity, Location Area identity, the minimum received signal level required to access the network, etc.)
  - **Frequency hopping**: if the association uplink/downlink lasts over time, rather than transmitting the entire message over the same channel, it slip in sub pieces sent in a pseudorandom sequence (sent to the MS), keeping switching the frequency used for each piece (Useful to make communication more reliable also over noisy channels).

- **Base Station Controller (BSC):** monitors and manages the resources of a group of BTS. From the BTS it receives the information about the quality over which we are transmitting. This is important because the MS is moving and the quality of the link can degrade, so is needed to allocate new resources in a novel cell for the MS and switch to a new BTS (***Handover***).
  - **BSC** allows the transcoding (we have different transmission from the GSM channels to the PCM channels) performed by the TRAU.

Network Switching Subsystem (NSS): It route information to/from the MS, connecting the MS to external networks, and also performs mobility support. It is made of a number of switching elements:

- **MSC**: which performs mobility management, associated with a VLR,
- **VLR**: that stores temporarily information about users currently located in a LA under the MSC, including the services that can be accessed.
- **HLR:** main database and maintains all the information about the user such as the IMSI, etc. It also stores temporary information as in the VLR, parameters for identification and encryption.
- **AuC**: it dynamically generate and transmit to the network the triples used for authentication and encryption of data to/from the users.
- **EIR**: database which contains identification and characteristics of GSM terminal equipment, together with the manufacturer, country of manufacture, etc. used to protect the network from the use of equipment stolen or not compliant to standard or malfunctioning terminals.

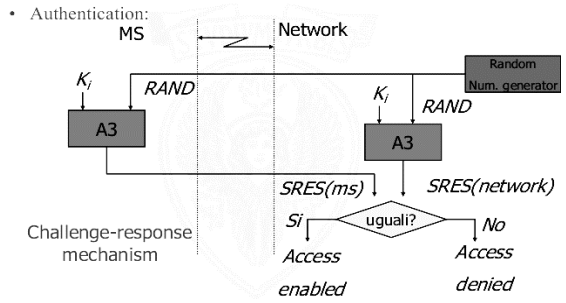


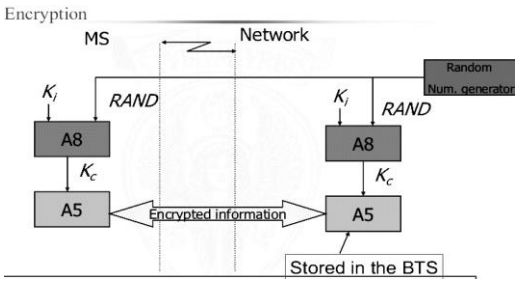
Additional **Operation and Maintenance Subsystem**: menages the components of the operator network (BTS, BSC, MSC...). Includes the functional units responsible for monitoring and control the network, its maintenance and remote management.

Security procedures of GSM:

1. **Authentication** to verify the user's identity: when a user wants to get access to the service, a *RAND* is transmitted through the operator network over a radio channel from BTS. Then, using A3 algorithm stored in the SIM, the K_i key and the *RAND*, the user produces an *SRES*, and the same is done on the network side. If the received SRES is the same as the one computed based on the information on the network, the user is authenticated.

2. **Encryption:** If it is authenticated, the RAND is used with the algorithm A5, based on the encryption key K_c (obtained using A8 algorithm), to encrypt a standard message which is transmitted over a control channel to the network that does exactly the same.

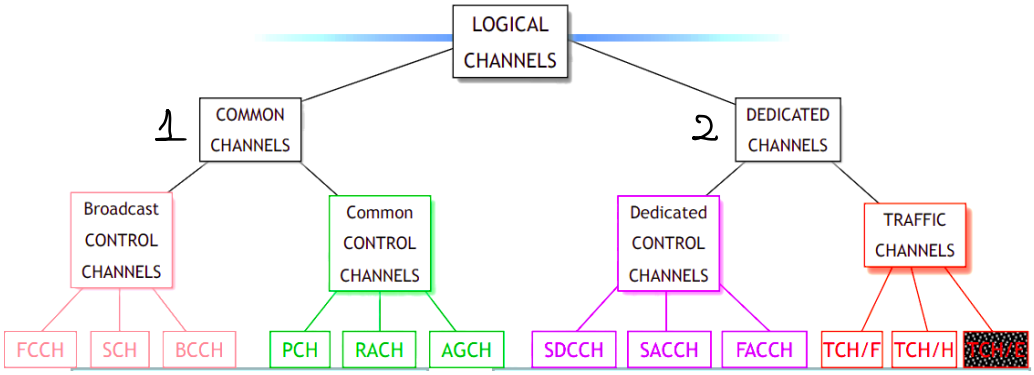




LOGICAL CHANNELS

There are different logical channels to provide services, to exchange information or traffic data. The Logical channels are divided into:

1) **Common channels:** it multiplexes information for several users (not assigned to 1 user).
2) **Dedicated channels:** channels assigned to a given user to exchange information



1) COMMON CHANNELS:

- **Broadcast Control Channels:** unidirectional downlink to exchange information about how to work in a cell (e.g. how to sync) in a cell addressed to all users in that cell. Divided into:
  - **FCCH** (**Frequency Correction Channel**): used to correct MS frequency, to allow the synchronization of the transmission with the BS.
  - **SCH** (**Synchronization Channel**): to receive info about the BS under which I'm located, after the frequency synchro.
  - **BCCH** (**Broadcast Control Channel**): carry general information that are broadcasted to all user of a BS (about channel structure, about the BTS and the neighbours, Location Area Code, etc.)

- **Common Control Channels:** channel in which the BTS can transmit info to/from a user but over time it also connected to all the users in that cell to receive information. Divided into:
  - **PCH** (**Paging Channel**): in which a *page message* is sent to localize a specific user in a cell. If the user is found it will answer with a RACH.
  - **RACH** (**Random Access Channel**): channel in uplink (MS →BTS), over which users request resources (e.g. to setup a call).
  - **AGCH** (**Access Grant Channel**): since multiple users can request resources at the same time, to avoid collision the AGCH is used as a response that the request has been correctly receive, including also the information of the channel I want to allocate. This channel will be used to perform the Authentication procedure, do the call setup etc.

***Slotted-ALOHA protocol:** when I request for resources and a collision occurs, I don't receive any ACK so I wait a random time and try again.

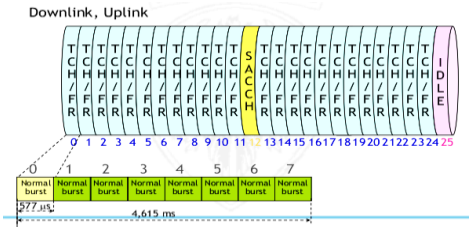
2) DEDICATED CHANNELS:

- **Dedicated Control Channels:** to carry information specific to a given connection, during the setup or while in conversation.
  - **SDCCH** (**Stand-alone Dedicated Control CHannel**): channel allocated at the phase of setup of a call.
  - **SAACH** (**Slow Associated Control CHannel**): while in conversation, is allocated to keep exchanging information with BTS (about the quality etc.)
    - in downlink: ***time advancement***, ***Power control***, frequency hopping sequence, frequencies used in other channels.
    - in uplink: FER of the downlink traffic channel.
  - **FACCH** (**Fast Associated Control Channel**): to fast contact the user which is conversation, and sending information about how to perform handover (new BTS and channel to use)

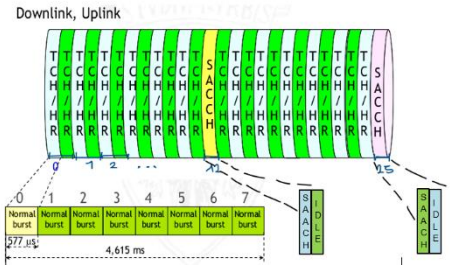
- **Traffic Channels:** channels which carry both voice and data, both in downlink and uplink.
  I'm able to transmit in a slot per frame over a carrier that changes according to pseudo random frequency.

To optimize the allocation of resources when we have limited resources as when transmitting over a wireless channel, the idea is to use the same physical resources to multiplex more logical channels. To give a representation of the mapping between logical-physical channels we use the **Multi-frame** concept: physically I get associated a slot per frame over a carrier frequency but if I want to see what is transmitted to the specific user, I can look only at slots related to that user.
A user get allocated:

- a full slot in case of full-rate encoding → **MULTIFRAME TCH FULL DUPLEX**: used to transmit traffic to/from the user but in the 13th frame there is a slot used to send SACCH information, so are multiplexed on the same physical resources two logical channels.
  also don't need a control frame every 13 frames, but every 24, so I can use the 25th frame for being idle.



- half of the resources in half-rate encoding → **MULTIFRAME TCH HALF DUPLEX:** in which is multiplexed two data traffic flows (and two SACCH) over the same physical resource. in this case, two users alternate for each frame. The 12th is dedicated to SACCH control info associated to user X, while the 26 is dedicated to SACCH control info of user Y.
  Cons: transmitting in half of the speed



When we transmit information over a physical channel, we transmit *packets*, which in GSM terminology are called **Burst** and define the use of the slot (how much used for info, how much for the guard time, or the packet structure etc.). due to TDMA each block should be transmitted with a given power to avoid interferences with other slots. So I have:

- **Normal Burst**: Used for user transmissions (speech or data) over traffic channels
- **Access Burst**: Used to transmit information over the RACH and for First-time access to sync and estimate the Timing Advance etc.

Different types of bursts:

- **Frequency Correction Burst** – Used over the FCCH to correct the frequency of the MS
  Synchronisation Burst: used over the SCCH to transmit information about sync for slots and frames
- **Dummy Burst**: pseudo-random sequence transmitted when I don't have any information to transmit over the BCCH because the user still a signal to estimate at which strength is receiving also from the adjacent BTS. (downlink only)

PROCEDURES

The user is identified by the IMSI and it can associate to the network and gets the information needed to operate with these procedure:

Call selection: the MS is able to scan the carrier frequencies and identify its carrier frequency and then it sync to the BTS, receiving some information (e.g. LAI) through the common channels.

Registration: After I receive the LAI, I need to inform cellular network operator system about my location LA, to maintain this information in database.

- If it is the same LA in which I was I perform an **IMSI attach** anyway because after some time the MSC/VRL sets a detached flag on me pointing that I'm currently unreachable.
- Otherwise, I have to perform **Location Update** to inform the network, and update all the database about my current LA.

I can be in two different situations when I move to another LA:

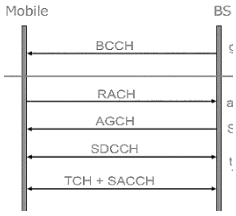
- Still remain under the control of the same MSC/VLR → **Intra MSC:** The MS send a Location Update request that goes from the BTS through the BSC and reaches the MSC/VLR which change the TMSI and store the new LA. Then security procedures are performed and it receive an ack to confirm the operation.

- I'm changing the MSC/VLR → **Inter MSC:** this time the request reaches the new MSC/VLR, which ask the IMSI of the user to the old VLR, and then ask to the Home Location Register whether that user can performs that request, and receives the answer. Then, after security procedures, the new VLR sends inform the HLR that it is the new VLR in charge of managing that user, and if it send an ack, the Location update is performed between the new VLR and the MS (as seen previously). At the end the HLR communicates to the old VLR to cancel all the information about the user.

Call procedure:

An MS can establish a call to another user by using the number, this allows the **PSTN** network to establish a setup of the call until the **GMSC (**Gateway Mobile Switching Center) of that cellular operator, which contacts the HLR that knows under which MSC/VLR is the user, and return the Mobile Station Roaming Number (MSRN). In this way determine how to route until the MSC/VLR.
The MSC/VLR knows the LA of the user but not the exact position, so it ask to the BTS managing the cells in the LA to send a paging message to find the MS. The MS will reply by sending a request over the RACH and then get allocated over a SDCCH to do security procedure and then to allocate a TCCH to establish the call.

Summary steps:



- BTS send info in broadcast over the BCCH to all the users,
- MS will make an access request to the BS over the RACH, it will answer back over the AGCH and get allocated to the SDCCH (to set up a call).
- after authentication etc., I'll get assigned a specific Traffic Channel + the associated SACCH and I can finally start exchanging data.

NB:

- When I transmit over a data channel I do it on a specific carrier frequency and time but it takes a very small time to receive a packet which depends on the position of the Ms relative to the BTS, so even if the MS and BTS are aligned I actually have moved. In addition the BTS will receive the message with a delay which is the propagation time so a collision could occurs if the transmission overlap with the transmission of the next slot. In this case I use the Guard Time, in which I ask the MS to not transmit for a portion of the slot, large enough to avoid overlapping. But if I use it I am not actually using all my capacity (lower data-rate transmission) → So I do **Timing Advance**: the BTS alert the MS that its transmission is coming with a delay, so that it can advance it and so the BTS can receive the transmission at the beginning of the slot. During all the transmission they exchange information to keep adjusting the time.

- ***Power control** in order to increasing/decreasing transmission power during the conversation.

- ***Interleaving** is used in order to improve overall quality and be able to correct errors with FEC because we are spreading the error among the transmission of multiple users.

MOBILITY MANAGEMENT

In cellular system we provide the service globally by implementing **frequencies reuse** (see cap.1). In general a smaller cell:
PRO: provide more capacity per user because there are less users that compete for resources. We can also use power control to save energy.
CONS: we have more BS, more devices and more need to cover the area (so additional infrastructure costs). Moreover we have to support *mobility management* to maintain the quality of the connection if the user is moving on the border and to do the handover if he pass to another cell, and repeat all the procedure seen before (reroute the setup of the call to the new BTS, update information etc.). If the user is moving in:

- IDLE state → perform Location Update, Cell selection
- ACTIVE state (while in conversation)→ handover

HANDOVER: procedure by which the MS in conversation moves into another cell so need to change its BS. It is performed by the BSC according to different strategies based on some measurements (received signal strength, quality, etc.) carried out by both the network (BTS) and the MS.

General reasons:
- quality of transmission is degrading
- if MS is moving out of the cell
- high density of users so a BTS is no longer able to serve all of them
- Control and maintenance reasons

Types of handover:
- Hard handover (GSM-2G): Removal and establishment of a new radio link
- Soft handover (UMTS-3G): the user is simultaneously connected to several BS

Recall that we require the received signal to be above a given threshold to be able to correctly receive the signal→ the choice of the threshold of activation of the handover procedure is a critical factor.

There are several methods to do the handover:

1. **of the strongest signal:** when I receive a stronger signal from another BS with respect to my actual one
2. **of the strongest signal with the threshold**: If the signal from the previous BS < threshold and the power of another BS is stronger
3. **of the strongest signal with hysteresis:** if the power of the other BS is stronger than a value of h;

PROBLEM→ when there is the handover, I release the old channel in the old cell and I allocate a new channel in the new cell, But the channel in the new cell may not be available.
We can define:

- **Pdrop** as the probability of rejecting an handover which should be 0. (e.g. temporary bad channel)
- **Pblock** as the probability of rejecting a request (call setup)

NB: it is better to block an incoming call that loosing one active.

There are techniques for recelling the user to ensure that when there is a request of handover we can support that.

- **Guard Channels:** a portion of the channels allocated to the cell is reserved to serve handover requests.
- **Queuing priority scheme**: when we measure that there is the possibility to perform a handover, start request resources to the new BTS, and if the channel is not available the request is buffered and served as soon as the channel is freed.
- **Subrating scheme:** If there are no channels available at the new BS a channel previously allocated to a call is divided into two channels each half rate.

The way methods are implemented and the way different elements of the cellular system operator network are involved depend on the specific situation:

- **Intra Cell - Intra BSC:** I'm in the same cell managed by my BS but I need to change channel due e.g. to low quality. The Handover is decided by the BSC only and is allocated a new traffic channel.

- **Inter Cell - Intra BSC:** when I move from a BTS to another one, both under the control of the same BSC. The handover procedure is fully controlled by the BSC, which identify the best BTS, decides the new channel and inform the BTS and the MS.

- **Inter Cell - Inter BSC:** when the BSC, due to measurements, decide that the best cell is another one of a BTS which is not managing. So the BSC, through the MSC reach the new BSC managing the new BTS asking to allocate a channel to that MS and take care of that.

- **Inter MSC:** when the BSC decide that the best cell is another one of a BTS which is not managing and is under another MSC/VLR. It send an handover command to its MSC that send the request to the new MSC/VLR which manage the new BTS. More complex because the call setup has to be re-routed to the new MSC/VRL.

VOICE CODING (+)

Digital signals must be appropriately modified in order to be transmitted. The modification of the signal for transmission is called signal encoding. There are different kind of encoding that have been proposed, and in general the quality perceived by user is measured according to **Mean Opinion Score (MOS)** and depend on the specific encoding we are using.

- **Waveform codecs:** to be able to reconstruct the signal we can sample the signal by splitting it into equal size intervals and approximate the sample values. more bits we use, the larger is the number of intervals so smaller is the possible error in the while reconstructing.
  we can also use a quantization, or we can use a predictor and then consider the difference between the predicted signal and the real one. performance improves if them are adaptive.

- **Source codecs (vocoders):** when we consider a voice signal we can develop some kind of coding using the information we have about human voice to compress the information we need and remove redundancy from the vocal segments.
  - **Linear Vocoder** (**LPC**) encode the parameters of the synthesis filter and the excitation sequence. In decoding a synthesizer uses the received parameters to reproduce the signal.
    We get a low bit rate but a sort of metallic voice → To overcome we can allow a wide variety of excitation signals in such a way to use a finite number of parameters to reproduce the voice signal, then use one of the codes seen before
  - Then I can use **Hybrid Codecs** in which, starting from the starting point, encode only the different signals.

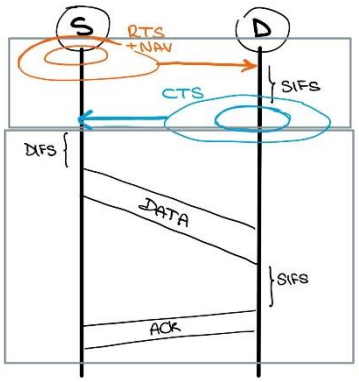
AD HOC NETWORK ROUTING

In an Ad hoc network we have several devices which can transmit/receive data and act as router for relay information to a final destination. This kind of devices must be: **self organized, self configured** and **self maintenance**.
These kinds of networks are used in military communications, disaster recovery scenario, IoT etc.
In IOT world the network s highly dynamic since devices are very resources constrained in terms of energy, memory and computation capabilities and can appears and disappears over time. The traffic is usually between sinks-sensors or only sensors.

Since devices send information over a broadcast channel, we need a **MAC PROTOCOL** to **avoid collisions**. Here we cannot use CSMA/CS because users cannot transmit and receive at same time). We don't use TDMA because synchronization or scheduling is not easy in highly dynamic network but in this way we cannot guarantee real-time (so sometimes we can consider hybrid solutions also to apply also TDMA).
The protocol developed is the **CSMA/CA** (Collision Avoidance).

The device performs carrier sensing before transmitting to check if someone else is already transmitting over the shared channel. If someone else is transmitting:

- If the channel is free after a time called **DIFS** the node can start transmit, otherwise the node waits for the end of current transmission + a **random backoff time** and counts down this value after DIFS. When the counter reaches 0 it can transmit.
- The packet has been successfully received by the destination, if the node receives an ACK after a **SIFS** time ($SIFS < DIFS$).



Moreover, to improve the transmission and avoid collisions, before sending data, an host sends a small control packet **RTS** (Request to send) in broadcast to all its neighbours to request the access at the channel. It contains a field called **NAV** that expresses the estimated time needed to send the data. When the destination receives the RTS, it waits for a SIFS and then reply with an ack called **CTS** (clear to send) in broadcast which authorizes the node to transmit and communicate to the other that for a certain time (NAV) the channel will be busy.

Upon receiving the data packet, the destination waits for a **SIFS** and then sends the **ACK** to communicate that now the channel is free.

The goals of **ROUTING** in ad hoc networks are:

- Multi-hop path routing capability, from a source to destination not directly connected.
- Dynamic topology maintenance, because of mobility or duty cycle.
- Avoid loops.
- Minimal control overhead.
- Low processing overhead, computation power of IoT devices not so high.
- Self-starting.

1) The first idea is to apply the same approaches used in traditional networks (internet): **Link State** and **Distance Vector.** This is what has been done by the so-called **Proactive Routing protocols** that exchange information among all the devices keeping updated the routing tables, before the sending of the packets.

- **Link- State:** which each router shares knowledge of its neighbours with every other router in the network to have at the end a complete view of the network topology (Information sharing takes place only whenever there is a change). Then the routing tables are computed by using Dijkstra Algorithm and then can be computed the shortest path.
  PRO: widely used in large networks due to their fast convergence and high reliability.
  CONS: in Ad hoc the dynamic topology led to a huge amount of control traffic information.

- **Distance Vector:** in which each router computes a distance (measured as number of traversed hops) between itself and each possible destination (i.e. its immediate neighbours) and exchange its information about the network to its neighbours to allow them to update the routing table. These tables are made by implementing Bellman-ford and the route with the least number of hops will be the best one.
  PRO: we don't need a global view but only local information. No syncro is needed.
  CONS: slow convergence and suffer from the ***count-to-infinity problem*.**

* **Count to infinity:** problem which arises because of *routing loop* in a network since the nodes will infinitely increase their estimate on distance metric. The loop occur when a packet is continually routed through the same routers over and over, in an endless circle, because for ex. a link or node failures render a node unreachable or two-routers send updates at the same time. Some solutions:

- *Bounded network diameter*: using a fixed diameter (e.g. max 15 hops)
- *Split horizon with poison reverse*: in which routing information is never sent from an interface from which it was learnt (prevents reverse routing) to avoid loop.
- *Trigger Updates*: update in case of network changes and not in a periodic way.

2) Another idea, to avoids these problems is based on **Highly Dynamic Destination-Sequenced Distance-Vector** (**DSDV**) **Routing** based on a smarter DV approach able to detect when a device disappears (or appears) from the network topology to avoid count-to-infinity problems.
   to do this the idea is to consider a new metric based on the idea of **sequence number** associated with "how fresh" our information is in computing the route. When a link break and the topology change the sequence number increases and each node recompute their distance vector and propagate the updates + the sequence number. A node will prefer the estimate with the highest (fresher) sequence number.

To support a highly dynamic network (i.e**. IOT scenario**) the DSDV protocol is evolved using **Optimized Link State Routing** (**OLSR**) which aim to <u>optimize the amount of control traffic exchanged in the network to flood the information with less energy resources.</u>
The idea is to reduce the number of re-transmission by using **multipoint relays**:
A node X selects its MPR(X) among its one-hop neighbours discovery (using "hello messages"), and only this set of multipoint relays of node X are allow the re-propagation to reach two-hop neighbours. In this way, instead of sending to all its neighbours it sends information only to the MPR(X). The other neighbours of X, not in MPR(X) receive and store the broadcast messages transmitted by X but do not retransmit them.
Then, each node locally runs a shortest path algorithm to determine paths to the different destination and creates a routing table.

- Different link metrics can be used in combination with OLSR → e.g. ETX which try to minimize the number of retransmissions and used power in wireless networks.

3) In IOT systems sometimes it is better to exchange information only if we need (event-based) and not continuously (requires a lot of energy consumption), so in this scenario is proposed a different kind of protocol called the **Reactive routing protocols**: which avoid exchanging control traffic and update routing tables continuously even if nothing happens.

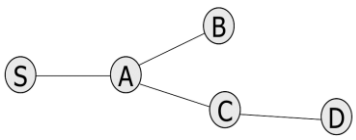
There are two examples of this kind of protocol:

1) Ad hoc On-Demand Distance Vector Routing (AODV)

This is the reactive approach of **DSDV**. It discovers possible paths through the **route discovery cycle** and only keeps information of **active routes**. If paths are not used for some time, they are discarded and the discovery cycle must be repeated. Nodes that are not on active routes do not retain any routing information nor do they participate in the periodic exchange of routing tables.

- If the network topology changes (e.g. broken link), a distributed approach is used to decide on new paths based on local information exchanged with neighbours.
- Since it is a Distance Vector approach, it may suffer from **counting to infinity** problem, which can be solved again by using the idea of sequence number.

Method strategy:



- Source S needs to route to destination D so it checks for a valid route but to discover it creates a specific control packet called **Route Request** (**RREQ**) to broadcast to its neighbours. In this packet it enters the destination address and sequence number regarding D, then it puts its address and its sequence number (of S) and a counter to count the number of hops.
- A receives the RREQ from S and reply, and it makes reverse route to S (updating count=1). Then it maintains information on the route discovery process, and checks if it has a valid route to D, otherwise it rebroadcast the RREQ to its neighbours (also to S).
- B and C both receive the request and note the reverse route and send back to S this information (updating count=2). Then they check for a valid route for D in the routing table.
- B re-propagates the Route Request but after some time doesn't receive a Route Reply so it cancel the entry for the reverse path.
- C has a valid entry for D (valid = fresher), and create a new control packet **Route Reply** (**RREP**) with the D address, its sequence number and the count, and send it to S using the reverse route through A.
- A receives the RREP and note that it has a good router to D and propagate in unicast towards S.
- When S receive this RREP update the routing table adding the good route to D.

    - If there are multiple valid routes we re-propagate to the source only if this Route Reply improves the estimate (> SN and < count).
    - If at a given point, one of these links can break we no longer have a valid route to D. In this case the node communicate to the active neighbour this problem, this is done using Route Error (RERR) Messages to all destinations. This is propagate back to S which delete that route.

Optimizations:

- **Expanding Ring Search:** prevents flooding of the network during route discovery whenever we don't have a valid route. Done by sending a Route Request with an extra field TTL (Time to Live) which forces the packet to propagate only one-hop (the idea is that maybe one of our neighbours has a valid route). When we don't receive the Route Replay the TTL increase to see if the neighbours have a valid route; then we repeat the same reasoning.
- **router maintenance:** repair breaks in active routes locally instead of notifying the source, send the REER with a small TTL

2) Dynamic Source Routing (DSR)

Here the Route discovery cycle is computed only if there is something to transmit and it use the *source routing*: keep information about the traversed nodes in the Route Request packet (in the field ***route record***) while propagation. It learns explicitly the route to follow and include this in the packet.

The sequence number is not used.

Optimizations:

- **cache overhearing minimization**: a node that receive a request/reply learns the intermediate routes and cache all these information. The idea is to use the cache information to discover better routes. A node that has a source route in cache can immediately reply to the RREQ.
- **RREQ overhead minimization**: to decrease the quantity of RREQ sent if the network is disconnected. The idea is that if a node Y that receives from a neighbour X a source route which is **Path shortening**; can report it back to X through a so- called "unsolicited RREP".

AODV and DSR Differences:

AODV	DSR
• uses next hop entry • uses route table • The route table entries do have lifetimes	• uses source routing • uses route cache • route cache entries do not have lifetimes

Limitations:

- **Proactive** → exchange a lot of control information even if we don't have to transmit,
- **Reactive** → route discovery whenever we want to transmit data (overhead and +latency)

Moreover, in an ad hoc network, there is a **scalability issues** because I need to find some way to have some information update on the change of position of the destination.

Geographically Enabled Routing

In some applications can happen that we would know extra information as **position** and **time**.
If we consider powerful devices e.g. with GPS, we can provide these extra information, and in this case we can perform other kind of strategies to route → **geographic routing**.

By using geographic information I can use the GPS or I can send a packet at least in the direction of the destination (if I know its position). In this way I don't even need to update a routing table over time but only my position as a potential relay, without knowing the network topology and so I don't care If the topology changes.

- The accuracy of the localization depends on the used technique to localize.
- It requires extra information so extra hardware or protocol to localize
- No scalability problem but anyway I need to know the destination so periodically it has to send update on its position.

NB: in IOT world the position of the destination (sync or AP) is static so I don't need it.

Location-Aided Routing (LAR)

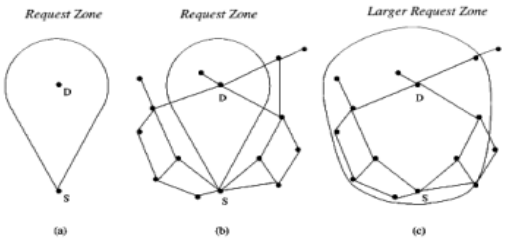


Figure 3. Request zone. An edge between two nodes means that they are neighbors.

The idea is to transmit over a "cone" area large enough to cover the expected zone in which the destination could be. This is because the destination that could be a MS is moving.

(- recent information about the position → + will be the cone).

In the destination is inside the cone I can reach the destination, otherwise I use a backup procedure in which if I don't have a valid route (case c and b) through the nodes inside the cone, the procedure envisions to enlarge the cone area.

- LAR was initially introduced to solve the inefficiency of reactive protocols to limit the route discovery over the network→ flooding directional with the cone scheme.

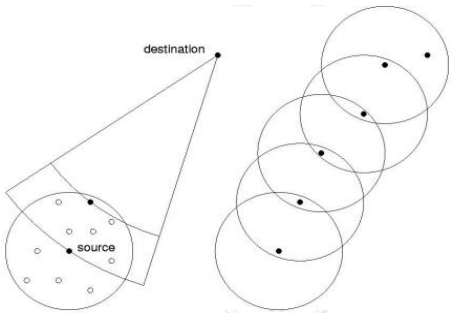
LAR – 1: same approach but propagate over a square including the source and the expected region (directional routing).

LAR – 2: It is setting the so called *greedy forwarding rule* for geographic routing. The idea is to find a valid path to reach an area close to the destination:

- Source know its position and the Destination position, and each node (potential rely) own position
- According to the greedy forwarding, a potential relay s good if it is in the ***positive region*** (positive advancement in the direction of the destination)

(a node is inside a positive region if it is inside a circle with radius equals to the distance S-D).

- Then pick the best relay in that positive region (=the closest to the destination)



DREAM

It is a directional routing (similar to LAR), proposed because if everything is mobile and everyone is transmitting, the fact that the destination may also proactively distribute its location information may be an issue.

The observation made in DREAM is that due to a "***distance effect***" the cone area will be large enough or not depending on how distant this neighbour will be. In this way nodes closer to me tend to go out faster than the one more distant. This means that I need to receive more location updates from the closer nodes.
→ the idea is that: in less time the node goes out of the region, the more often it should be send to me the updates.
This idea reduces the overhead in the network.

Let's consider now an **IoT scenario**, in which we assume the topology will change because nodes duty. Consider a geographic routing to also decrease the energy consumption.

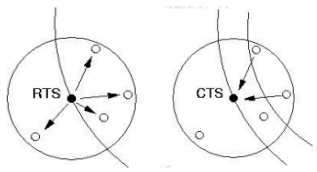
- **Geographic routing**: each node needs to know its location, the destination location, and the location of whom is transmitting (communicated in the packet)
- **Greedy approach**: tries to select relays so to advance as much as possible toward the destination at each hop

→We have a problem to address to **avoid collisions**.

The first idea is that I could go from the approach in which we have both **MAC** and **Routing** to the so-called **cross layer approach**, where the two aspects are jointly managed. →**GeRaF**

GeRaF:

It is a **cross-layering protocol** implementing the **greedy forwarding** rule:

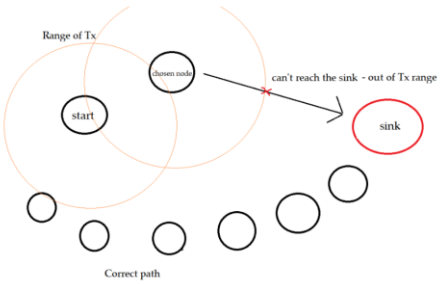


I am a black node, to learn who is the best neighbour, I send an RTS (including my position, destination position etc.).
Each neighbour compute if it is in the positive region and if it is become a black node and could answer back with a CTS.

If more than one answer back → **collision**!

- In case of collision of the CTS, the idea is to minimize the problem **splitting** this set into subregion, query with a RTS starting from the region closer to the destination (first region), and if no one answer back decrease into a second region, and so on.
- Then If I cannot detect the packet but receive some noise, I can send the **collision packet** (in addition to RTS and CTS) to that region. If I am the node that was expecting data but receive the collision packet, I toss a coin and with probability p I answer back with the CTS. If no one send the CTS, another extraction has to be done. If all areas are polled unsuccessfully we have to try again after some time decided using an exponential back-off.
- In some cases nodes can **desynchronize** their CTS response by computing a jitter based on their quality to advantage better nodes over worst one.

Problem: GeRaF does not ensure that we will always reach the destination because of so-called "connectivity holds". This problem happens often in low density networks.



←e.g. if I apply the greedy forwarding rule I provide advanced node but I don't obtain a valid route to reach the sink. In this case I have a **dead-end**. To solve I need to go backword and find a valid route.
To avoid this problem I can set a fixed number of attempts to find a relay and every time a node fails to find a relay it decreases its duty cycle. Over time

nodes that lead to dead-ends are less and less selected.

The assumption we have made to solve the dead-end problem is not realistic and failed at test. → without this ideal assumption, you are able to deliver according to this rule 40-50% of packets → unapplicable. You need to find other solutions → ***ALBA***

ALBA

ALBA is a geo routing which tries to solve practically the **dead end problem** using the *cross layering approach* specifying an objective function to choose the best next-hop relay based on how effective a relay is able to receive and correctly forward packets.

Different metrics are used to perform this choice:

- **GPI** (Geographic Priority Index): depends on how close a relay is to the final destination.
- **QPI** (Quality Priority Index): indicates how effective is a relay and it's computes as follows:

$$QPI = \left(\frac{N\ packets\ in\ the\ queque + N\ packets\ to\ send}{\text{average length of burst expects to send correctly}}\right) - 1$$

- The advantage of using a cross-layer approach is that now every information exchanged can be used to optimize the transmission.

- **Collisions** can be avoid similarly to GeRaF, in which if a node receive a **collision message** then will send a CTS with a probability p.

- To solve dead-end problem is used the **Rainbow Mechanism**: it is a node colouring algorithm which works by recursive let nodes that are dead ends to stop volunteering as relays (without removing them completely as before!)
  all nodes start yellow, if a node can forward the packet became red and if it can't reach another red or yellow node become blue, then if can't again become violet. Each node can reach another node of the same actual colour of previous one.
  After some time each node stabilize to a colour and we are able to route the packet without dead-ends.

Location in Sensor Networks

In a sensor network we seek:

- For solutions which overcome **scalability issue**,
- Which are **energy saving**, **simple** and well integrated with **awake/asleep** schedules
- Which do not require to maintain routing tables

When we use a **geographical routing** protocol is needed to implement ***localization***, whenever is not possible to use GPS, this increases power consumption and works only in case of LOS. To enable localization, we need to be able to compute the distance ("**ranging**") w.r.t. reference points ("**anchors**") which are devices which position is known.

1) **Absolute Localization:** at the beginning only few positions are known, then by using **ranging** we can estimate the positions of other nodes, after that they propagate their position information to act as reference points. There is a localization error that is propagated. If the density of the anchor nodes is not enough this error is amplificated.
These techniques depends on the hardware capabilities of the devices:

- **AoA:** to determine the position of a MS, we estimate the angle of the received signal, and by knowing the positions of the anchor points, we can compute its location in 2D by using directive antennas and a compass.

- If it is not possible to estimate the Angle of Arrival (AoA), its possible to estimate the range by using the **received signal strength** (**RSS**) or the **time of arrival** (**ToA**).
    - in **ToA**, to correctly implement this approach the Tx has to timestamp the packets immediately before the transmission, and immediately after reception (and so nodes need to be synchronized); or the Rx send an ack to the Tx round trip time (no synchronized); or can be sent at the same time two kind of signal with different propagation speed and estimate then their difference in propagating time.
    - In **RSS**, ranging can be estimated using the received signal strength knowing the Tx power because we know that the received signal strength decrease in function to distance.
      (imprecise estimate due to multipath fading).

2) **Range free localization:** it can be used when is not possible to get an accurate localization using ranging techniques. It only uses already available information to get a ruff estimate of the position of the nodes.
here we assume the location of anchors as known, then the shortest path from each node and anchors is computed, and then the anchors estimate the shortest paths from each pair of anchor. At the end an average distance to pair of hops is used to estimate the distance (the precision depends on the density of the topology).

Low Power MAC protocols

In general, when we think about a possible **MAC protocol** we need to consider the **duty cycle** and the **awake/asleep schedule** of the devices trying to minimize the **energy waste**, which occurs in collision, idle listening etc.
In general, we consider two kind of MAC protocols:

1) **Synchronous**: nodes are synchronized by sharing their own schedule,
2) **Asynchronous**: no synchronization because the network is dynamic and can require more or less awake times. Moreover the synchronization is not easy to achieve with IOT devices.

S-MAC:

In S-MAC nodes share a ***synchronized* awake/asleep schedule,** periodically, so that each node always transmits and receives at a given time, avoiding energy waste (but introduce latency because we wait the receiver to wake up). In S-MAC there is a need for an initial set up period and time lapses where one node is listening to receive the schedule from the others:

- If X receive a SYN control packet from a node Y, it synchronizes with its schedule becoming a ***follower***, and then re-broadcast the schedule after a random time. In this way all the nodes with that schedule are synchronized and cannot collide.
- Otherwise, it goes to sleep for a random time T, then wake up again and send this value T to its neighbours in a SYN packet, becoming a ***synchronizer***.
- It can happen, mostly in large network, mostly at nodes that act as gateway (**border nodes)** that a follower receive multiple schedule to follow. In this case they follow both, consuming more energy.

Each node also maintains a schedule table that keeps the schedules of all known neighbors.

- The exchange of data between X and Y is done following **CSMA/CA** and a back-to-back packet transmission for data bursts.
- To maintain **synchronization**, update schedules and cope with the clock drift problem are sent periodically SYNs.
- Communication is done only if both devices are ON (active time), then **throughput is limited**
- But when a node want to send a packets it has to wait the relay to go on, **increased latency** and **energy waste**

NB: a fixed Active time could not be a good strategy because may be situations where you do not have to transmit and therefore spend Active time unnecessarily, or vice versa, you have too many things to transmit and the Active time is not enough → **TMAC**.

T-MAC:

T-MAC is based on the same features as S-MAC, but the Active time change dynamically adapting on the traffic level of the network(it lengthens if there is communication and decreases if the state is idle).
Unlike S-MAC-if no transmissions are received from neighbours for a time **TA**, the active time is aborted and the nodes go to sleep. TA is chosen according to some other time intervals.

When a node send a RTS but not receive a CTS could happen that:

- RTS not received due to a collision → wrong to reduce the active time → a node should retry to resend the RTS at least twice before going to sleep.
- destination node having a **full buffer** → it could reply with a RTS (instead of CTS), so that it acquire the channel to send the data "stealing" the priority, in such a way to empty its buffer (**buffer priority**).
- the receiving node gone to sleep earlier → **early sleep problem**: when if I have $A \dashrightarrow B \dashrightarrow C \dashrightarrow D$, node D goes to sleep much earlier, before C can send an RTS to it (may degrade throughput). To avoid early sleeping: when C receive an RTS from B and know that It will send data to D, C should notify D with a fake RTS (**FRTS)** that there will be traffic for it so that it stay awake.

B-MAC:

It is an asynchronous MAC protocol with several objectives (effective CA, simple implementation, scalability etc.)

- To have an effective collision avoidance it use the so called **Clear Channel Assessment** (**CCA**): it propose a way to determine if a channel is free by estimating the channel noise, taking some samples and checking if they are below the average noise level.
- no SYN packets are sent to sync duty cycles, but is used the so-called **Low Power Listening (LPL)**:
  - if a node wants to send data it sends a *preamble* before
  - when a node wakes up, turn on the radio and checks for an activity for a time=length of the preamble, if there is it increases power and stay awake for the required time to receive the packets, after which it goes back to sleep;

The energy consumed in transmitting will be higher due to the long preamble, adding extra latency because neighbours waits until the end of it.

X-MAC:

It solves some problems of B-MAC due to the long preamble, proposing shorter preambles (**strobed preambles**) with target address information (IP) on each. They are send in a consecutive way so that in the pauses between that the destination can send a fast ACK and the data can be exchange. This drastically reduce wasted energy and in terms of reliability, wasted resources are avoided

WiseMAC:

It is based on the idea to mix the strengths of synchronous and asynchronous approaches. We don't need to keep the nodes synchronized and all these preambles when we send data. The idea is instead to inform our neighbours on our duty cycle to tell them when the next time will be to wake up. The following transmission will have a small guard period to ensure that our neighbours will wake up and then we transmit straight away.
In this way we solve the clock drift problem of synchronous protocols in which the devices slowly tend to misalign their time.

IoT Standardization

6loWPAN

It is a low energy protocol used in IOT world. The idea is that IOT devices have to be able to connect with Internet (by using the sink as a gateway) taking into account that we need an optimization of how packets are transmitted over the protocol stack. To implement this we have:

1) **Physical layer** in which we have IEEE 802.15.4 (dedicated frequencies on which Transmit)
2) **Data Link layer**: we have MAC protocols defined by IEEE 802.15.4.
3) **An adaptation layer** called LoWPAN, to optimize and compress headers to make addressable the smart IoT devices with IP (no TCP/IP protocol because too energy consuming)
4) **Next layers**: having made an adaptation we use IPv6 (instead of IP)
   We will use an optimized UDP/IP protocol that allows end-to-end communication with traditional non-IoT IP devices.

1) <u>Physical Layer:</u>

- we use different frequency bands according to different geographical areas,
- we have a PDU structured as:

- As topology structure we could have star, tree or mesh because we don't need many hops, and in each topology we have several elements that co-exist:
  - *end devices*: terminals that can also route information/data
  - *PAN coordinator*: control nodes that deal with how to communicate in the network, which is elected when the network is started
  - *Co-ordinator*: support the PAN tasks

2) <u>Data Link Layer</u>: At the MAC level, there are two types of mode to handle collisions:

- **beaconless mode** → we use CSMA/CA
- **beacon mode** → we use a central point as coordinator, sending periodically a network beacon after which starts a *contention access period*, in which it allocates resource slots for communication between users and then a *contention free period*

3) <u>Adaptation layer 6LoWPAN:</u> To make smart objects IP addressable because they are identified by EUI-64 (8 bytes), while IPv6 addresses are longer (128 bytes). To do this we need to optimize and compress headers. In general both are too long so the idea is to allocate a local address (16bits) as part of our network and the rest as a shared sub-net mask (since the full address will be used only by the edge router).

- **Fragmentation:** used when transmitting PDUs at layer two and three larger than 128 bytes.

- We could have two kind of **routing**:
  - Route over forwarding: this uses the layer 2 addresses to forward data packets.
  - Mesh under forwarding: this uses the layer 3 addresses (IP).

- **Header Compression:** to compress as much as possible the IPv6 header. To do this are used:
  1) **HC1** – for IPv6 header: try to avoid shallow information (such as the version) without loosing information so that the edge router is able to reconstruct the original header.
     We use:
     - a bit C= 0 or 1 whenever the flow class information is significant or not,
     - NH bits to indicate if the next header can be skip or not,
     - a final bit to indicate if we have a HC2 header too.

  2) **HC2** – for UDP header: also the source and destination ports are compressed, so in a field of the HC2 header is given a subset of possible ports from which to make the selection.

IEEE 802.15.4

It is the standard for physical and MAC layers for IoT. It is good for some application but doesn't fulfill the needs of some emerging industrial needs that demands for guarantee on communication delay, no resilience on interference and it is not ideal in high traffic scenario.

In order to achieve these needs IEE 802.15.4 has released an **extension** in 2016 which brought some improvements such as:

- *Low Energy* (LE): to operate in low energy duty cycle.
- *Enhanced Beacons* (EB): they give the opportunity to design beacon messages that
- are application-specific.
- *Multi-purpose Frame*: flexible frame element.
- *MAC Performance Metrics*: that allow link quality calculation.
- *Fast Association* procedure

There are some variants:

- **BLINK**: it supports very simple applications where we simply have to provide an ID for identification, location and tracking.
- **AMCA:** supports dynamic multi-channel use in distributed beaconless networks.
- **DSME:** supports time-critical application for large networks with beacon PANs.
- **LLDN:** designed to support commercial application requiring low and deterministic latency in which many sensors/actuators monitor and control an operation.
- **TSCH:** designed to support industry applications by combining slotted access, multi-channel support and frequency hopping. It is topology independent and supports increased network capacity, high reliability and predictable latency, while enabling low duty cycling.
  To do this, the available channels are divided into time slots and each communication is mapped onto a channel slot and I can dynamically allocate resources to adapt to the needs and state of the network.

LORA TECHNOLOGY

LoRa is one of the most used technologies in IoT world or in general in **Low Power Wide Area Network** (LP-WAN) in which many little and efficient sensors communicate in ultra-low power over ultra-long distances. This situation brings a very high attenuation of the signal at the receiver, interferences, and the CSMA concept doesn't work anymore because the cell area is too big and can lead to hidden node problem.

- **PHY Layer:**
  to reach very far distances, one can either increase the transmission power (but it depends on the lifetime of the device) or use **spread spectrum technology** in which the transmitted signal is modified by using codes into it to increase the bandwidth of signal transmission, and thereby reduces the effects of interference, noise, and signal fading.
  At the receiving end, the codes are used to de-spread the signals for retrieving the original data. This increases the security of signal transmission.

- **MAC layer: LoRaWAN protocol**
  It supports:
  - secure bi-directional communications
  - mobility
  - localization

It defines 3 classes of devices:

- **Class A**→ perform uplink communications followed by two short downlink receiving windows. Battery-powered sensors and are energy-efficient,
- **Class B**→ like A, but extra receive windows at scheduled.
  Battery-powered actuators and are energy-efficient, while still having mechanisms for controlling downlink latency and using slotted communications that are synchronised with a beacon,
- **Class C**→ those powered by energy centres and are devices that can listen continuously (continuous receive windows), without having latency for downlink communication.

The network has a centralised intelligence that also allows low-cost gateways, as decisions such as identification, validation and localisation are made by the servers.

There are 2 layers for **SECURITY**:

- **Network**: by authenticating users and applying message integrity check.
- **Application**: separating application data from network operators.

And you can have two kind con security configuration:

- ***static attivation***
- ***over the air activation (OTAA):*** a node must perform a join procedure before sending data with the Network Server.

LoRa is a pure **ALOHA** system, there is no synchronization, a node sends a packet immediately if a duty cycle is satisfied. In case of collisions there will be a reschedule or cancellation.

Spreading Factor: as the factor that controls the speed of data transmission.
Lower the SF the higher the transmission rate is and lower is the time on air of the packets.
Normally, modulated signals with different SFs can transmit over the same frequency channel because they do not interfere. But in case of interference, LoRa solve it by taking only the signal with

the highest strength (capture effect).
- SF is also an important factor related to battery life (longer the active time-shorter the battery life).

To exploit the LoRaWAN it is essential to design suitable **allocation schemes** for the wireless resources. To do this are presented strategies for allocation of SFs:

- **Adaptive Data Rate – ADR:** based on the idea that if we reduce the SF we can increase the wait and reduce the TimeOn Air and viceversa. So a network server can indicate to and end device that it should reduce transmission power or increase data rate (end devices closer to gateways should use lower SF and higher data rate and viceversa).

- **EXPLoRa:** is an algorithm using *Sequential Waterfilling*, which sorts users according to their RSSI and allocates SF proportionally, in order to balance them. it aims to equalize the Air-Time channel usage for each group of devices using the same SF.

Collection Tree Protocol (CTP)

It is a Distance Vector protocol in which starting from a mesh topology we build a **tree** representation in which each device choose a ***parent*** to forward to base on some metrics as the distance to the sink or the quality over the local channel. each parent acts as a relay ”pushing” the node towards the sink.

- The metrics used to select the parent is the **ETX** = Expected Number of Transmissions to reach the sink, calculated w.r.t. the performance given by the exchange of message beacons. In this way you can see also the latency and the energy consumption this selection will have. The parent selection is performed only among non-congested nodes.
- In this protocol we have two kinds of packets: *Data Frame* e *Routing Frame* in which we have a “pull flag” and a “congested flag”, the sequence number, a time has lived value and the estimated EXT to also check if it goes back into a loop. If there is a **loop,** we force to recompute the routes.

These two things are implemented by the *Control Plane* and the *Data Plane*.

To check the **routing consistency** is used a validation criterion that simply checks if the $ETX(n_i) > ETX(n_{i+1})$. When this does not happen, a signal is sent to the control plane and the tree is rebuilt from the beacon frames.

To keep the topology update an high number of beacons are exchanged and to optimize this situation we can use the **Trickle algorithm** in which the frequency of beacons exchange depending on the stability of the channel I’m using, and it reset if we re-build the tree.

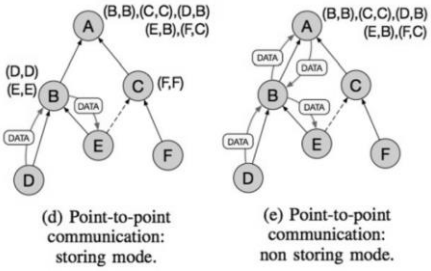
Routing Protocol for Low-Power Networks – RPL:

The approach is proactive and uses distance vector routing. It specifies how the network manager build a destination by building an oriented acyclic graph (**DODAG**) rooted in the edge router, based on the combination of different metrics depending on the traffic flows in the network (e.g. selecting the best ETX or avoiding battery-dependent nodes) and thus according to need.

To build the graph, the root sends a ***DIO*** *message*, which is then forwarded or not (if a leaf node) by its neighbours. Then node whose joint the graph send a ***DAO*** *message* to its parent which forward it to the root.
DIO messages are sent after a selectable timer or when an inconsistency is detected.

We have two kinds of communications:

1) **Storing mode**: data can be sent to the first ancestor with information on the destination, who then sends it to it.
2) **Non-storing mode**: data must be sent up in the tree until it reach the root node, which know how to forward to the destination.



EH-WSNs

One of the main problems of IoT networks is **energy consumption**: devices need to be operational for very long periods of time but are batteries- dependent and communication is expensive.
One approach has been to use **duty cycle**, which, using control mechanisms, makes the device go into active or sleeping states depending on whether data needs to be transmitted or not. In general, we seek a trade-off between energy consumption and latency.

→The new idea is to try to use **environmental energy** to power devices when we are in low power. This process of energy storage is called **Energy Harvesting**.

Obviously, since this kind of energy is unmanageable there is a need for a prediction of the energy that can actually be harvested to mitigate the uncertainty in energy availability. To do this, planning methods are used by proactively allocating energy using what is available, minimising situations where there may not be enough (and thus losing urgent tasks) and minimising energy wastage in the event of excess.

- **PRO Energy**: based on plotting various energy profiles by observing a number of typical days, and then being able to predict in the immediate future what the energy intake will be. In this way we can see which of these is the closest to the current profile, obviously taking into account how relevant and correlated what we have measured in the past is to what we will experience in the future (done by the *Medium term energy prediction*).

Harversting-aware routing can also be used, whereby the selection of the next hop also takes into account how much current energy is added to the predicted energy intake, so that higher priority can be given to those who are experiencing/will experience energy peaks.

NB-IoT:

In the infrastructure there were pre-existing devices that do not require so much different from normal wireless (connectivity, simple functions, low power consumption and so on).

So the idea is to repurposed and modified these existing technologies for cellular systems to reflect the needs of IoT networks (such as LTE).

NB-IoT are the kind of devices that transmit once in a while, often installed in places with poor coverage and/or no energy support (thus enforcing some requirements).

The objectives of NB-IoT are basically:

- minimise signalling overhead, given their low activity;
- use slow duty cycles (to spend little time on)
- simple functions and fast processing of data (always to be less on)
- appropriate security

Furthermore, only cell reselection in idle state is supported, there is a lack of handover and most of the advanced LTE features are not supported.

in terms of performance, NB-IoT is slightly more energy consuming compared to network such as LoRaWan, but still very good.

Cellular Systems - Beyond GSM

2G+ edge

Born when people start to need to transmit **data** as well as **voice**, so to support packet switching, high data rates, and higher quality of data encoding, as well as the need to extend the capabilities of mobile operators to enable them to provide more complex services.

- **full rate** codec based on **ACELP** for encoding/decoding
- introduced an **adaptive modulation and codec,** to reduce/increase the data-rate based on the Signal-Noise-Ratio of the channel.
- **Tandem Free Operation** that prefers multiplexing of flows of coded speech signal over transcoding (that degrades voice quality)
- The improvement of the physical layer brought also **enhanced data rates** and the possibility to allocate **multiple slots** to a single MS.

Moreover it performs:

- **Triangolarization** to localize an MS in a range of 100m from the BTS.
- **Number portability**
- **SIM** can ask to MS to perform some actions (reach the web, set a call …)
- **Cordless telephones**

- The need to **increase the data rate** is solved with **EDGE** as the radio access technology by changing the modulation phase (3 times more bits per symbol), without changing the system architecture.

- The raise of Internet also raised the need to support data transmission (not only voice) on cellular systems. The first technology used **GPRS**, which uses an IP backbone for packet switching integrated with the circuit switching networks. This change introduced **new kind of nodes**:
  - **SGSN** (Serving GPRS Support Node): route IP packets from/to a set of MS;
  - **GGSN** (Gateway GPRS Support Node): interfaces the cellular network with external packet data networks.
  - **PCU** (Packet Control Unit): new entity to manage data transmission over the radio channel

- When I want to establish a data exchange, I have to do a **PDP context activation** with the **GGSN** through which I receive an IP address (via DHCP) and through the sources request is exchanged.

- **Packet Data Channel** (**PDCH**) is a physical channel allocated for data transmission. Its structure is the same mix of FDMA and TDMA seen in GSM. PDCH is allocated only for the time needed to transmit the data and then released.
  Several MS can share the same radio channel, so several PDCHs are multiplexed on the same physical resource. The source allocation is done by using a Temporary Flow Block (TBF), required by the MS to communicate, which has associated a Temporary Flow Identity (TFI).

- Moreover, we do not have a distributed environment but a *central control point*, we can do some sort of flow scheduling, trying to optimise the use of resources by multiplexing users by using some new **Control Packages** over the Control Channels.

- Another change is on the **location** information. Now a new concept is introduced called **Routing Area.** Now, if a node is ***idle*** we know the Location Area (as in GSM), otherwise:

- When the data exchange request procedure starts the node switches to ***ready*** state and the location is kept at the Cell level.
- If the ready state has expired, because of inactivity, it switches to the ***Standby*** state in which information is kept at a level between LA and Cell, which is the Routing Area.
- When a ***standby*** timer expires, the MS returns to ***idle*.**

3G

The goals are to overcome the bandwidth of 2G+, to give full support to a variety of services and multimedia applications and the integration of mobile and satellite communications.

- **hierarchical** organization of cells: *Macrocell>Microcell> Picocell* (Smaller cells = better management)
- The **bandwidth** was **increased** to achieve higher data rate, changing also the amount of bandwidth provider per user.
- **TDD** and **FDD** (Time and Frequency Division Duplex) are used to divide resources between uplink and downlink, that now have also a different rata rate.
- **UTRAN** is used as the wireless access part and which:
  - monitors cell capacity and interference to optimize wireless interface resources usage
  - includes power control, handover, **packet scheduling**, call **admission control** and **load control** algorithms
  - radio channel encryption
  - congestion control to handle network overload situations
  - system information broadcasting
  - macro and micro diversity

- 3G uses **CDMA (**Code Division Multiple Access) as medium Access: in which at each user is assigned a unique **chipping code** (to encrypt/decrypt) in orthogonal way so that different users can send data simultaneously on the same physical channel without interferences.

4G - LTE

Due to the increase of Internet usage and devices, the main requirement is to increase the data rate even more, and so to improve the spectral efficiency.
Moreover, we have:

- Flexibility of spectrum usage
- reduced delays
- low energy consumption at the Mobile
- simplified network architecture

To improve the data rate we can't operate too much on increasing the bandwidth, because the radio spectrum is limited and most of the frequencies are already allocated, so the only thing to do was to improve on the spectral efficiency side by using the **LTE**.

- LTE supports:
  - **multicarrier technology**: **OFDMA** on downlink and **SC-FDMA** on uplink, so that greater flexibility, adaptability and robustness is achieved by using receivers with less complexity,
  - a single terminal can have **multiple antennas** for communication,
  - **packet switching**

- The **architecture** of the networks is:
  user equipment (**UE**) → **E-UTRAN** → **EPC** (Packet Core network) → **external networks**

- The BTS/BSC concept goes away and it's substituted by the **enhanced NodeB** (eNB), which ensures the necessary QoS for a *bearer* (IP packet flow with a certain QoS, passing between gateway and UE). It allocates downlink or uplink radio resources based on: the downlink data buffered, on Buffer Status Reports (BSRs) from the UE and based on channel quality.

- Inside the eNB there is a ***scheduler*** unit which distributes the available radio resources in one cell among the UEs. The scheduling can be Dynamic, based on feedback and QoS, or Persistent when allocated for a long time.

- **LTE Protocol Stack**:
  - **PDCP** - performs IP header compression
  - **RLC protocol** - responsible for segmentation of the PDU, and error correction by using ARQ method
  - **MAC** layer for scheduling and multiplexing the data according to priorities
  - **PHY** layer to performs coding, modulation, antenna and resource mapping.

Modulation scheme and code rate are <u>dynamically</u> selected based on predicted channel conditions provided as feedback indicator by the UEs (Channel Quality Indicator).

- In the **core network**, to simplify the architecture and management there is a separation between the elements of the *control plane* and those of the *user plane*:
  - **S-GW** - handles the transport of IP data traffic
  - **P-GW** - deals with IP addr allocation for the UE
  - **MME** - handles mobility management and security management
  - **HSS** - manages user data, the PDNs to which they can connect, the identity of the MME to which they are connected and is also the authentication centre (HLR + AuC)

5G

The factors that have led to the need for 5G are basically:
- greater high-speed connectivity,
- cheaper computing power,
- change in the type of traffic that is slowly becoming cloud-based and more distributed, and also the movement of intelligence components closer and closer to the end user.

The goal is to provide: a <u>delay critical, ultra-reliable, dependable, and secure broadband communications service to mobile users.</u>

The 5G provides:
- more available throughput,
- more speed to end users,
- lower latency,
- increase of energy efficiency, resulting in a better lifetime for low power devices
- huge coverage

It is not only destined to humans but to billions of smart objects in the emerging IoT network.

System level challenges:

- the different classes of QoS that the network must handle (in terms of throughput, latency, resilience, etc.)
- provide services across different infrastructures with different networks coexisting (**interworking**),
- density challenge to handled the increasing number of connected devices (e.g. IOT)
- diversity challenge to support the increasing diversity wireless solutions and traffic variety
- exploit every possible communication capability to optimise communications
- exploit energy harvesting in order to increase lifetime
- provide **privacy** and **security**
- mobility challenge due to discontinuous mobility between networks/technologies
- challenge concerning the accuracy of device **location**
- make communication systems **robust** to attacks and natural disasters
- improve the resource management (control, policy, protocols etc)
- truly flexible devices and control/protocol mechanisms to reallocate resources

- It also requires a change in the economic model, thus having to lower prices, while continuing to explore technological evolution.