Internet of Things

Simone Zannini

May 22, 2022

1 Wireless Systems

1.1 Wireless vs. wired

Wireless networks have a lot of differences with respect to the *classic* TCP/IP wired networks. The unique features of the transmission medium have a big impact on the network design, due to the involvement of issues such as low reliability, broadcast feature etc...
Besides, wireless systems are designed to work anywhere at anytime, thus we have to support mobility and portability, also concerning about external power resources.
The **broadcast** feature enable a wireless device to be heard by all other devices in its *transmission radius*. This feature doesn't come with no issues, but raises security challenges. There is the need of a **shared channel** and thus we have to control the interaction with the transmission medium (e.g., via MAC). The limited resources must be shared among the users.
Wireless communication comes with an intrinsic **higher bit error rate** w.r.t. wired one. Hence we have to implement mechanisms of error detection, correction along with re-transmission techniques.
Another crucial aspect is that wireless system often rely on **external energy sources**, thus there is a need of designing low power platforms and protocols that are energy efficient. This challenge brings out the computation vs. communication trade-off, that can be solved for example via hardware techniques that limits energy consumption.

1.2 Wireless Systems Models

We have **infra-structured networks** like the Internet, in which the communication is between a mobile user to an access point and vice-versa and **Ad Hoc Wireless Networks** that are based on peer to peer communication in which each node acts either as source/destination or as a relay.

1.3 Transmission Errors

With **BER** we refer to bit error rate. Like we said before it is a lot higher w.r.t. wired networks and this is because of issues like the attenuation, reflection and diffraction of the signal. Broadcast channel thus have the need to be regulated by a MAC protocol; antennas cannot transmit and receive simultaneously, we are able to use carrier sense and do collision detection.
The **hidden terminal** problem raises when two nodes A and B transmit a packet to the same node C. Neither A nor B can directly spot the collision.
We use **routing** for mobility and dynamicity in order to find the best path between nodes.
One of the biggest challenges in ad-hoc networks is how to minimize energy consumption considering the variety of nodes residual energy. Every layer of the protocol stack requires to find an efficient solution.

1.4 Wireless channel

Much less reliable than wired channels because the signal, while propagating, can face:

- Attenuation as function of the distance between transmitter and receiver.
- Attenuation due to obstacles.
- Propagation over multiple paths (problem of the multipath fading).

1.5 Radio signal propagation

We have different ways to refer to diffrent situations that we can encounter when looking at signal propagation:

- **Line of sight:** is when the signal transmits without encounter any problem. Non necessarily existing.
- **Reflection:** is when the signal bounces on a surface and then reaches the receiver.
- **Shadowing:** is when the signal is completed blocked by an obstacle.
- **Diffraction:** is when the signal encounters a surface with sharp edges that bends the wave.
- **Scattering:** is when the signal encounters objects smaller than the wavelength resulting in a multiple propagation of the wave.

1.6 Radio signal attenuation

Which law expresses signal attenuation as a function of the traversed distance? We assume

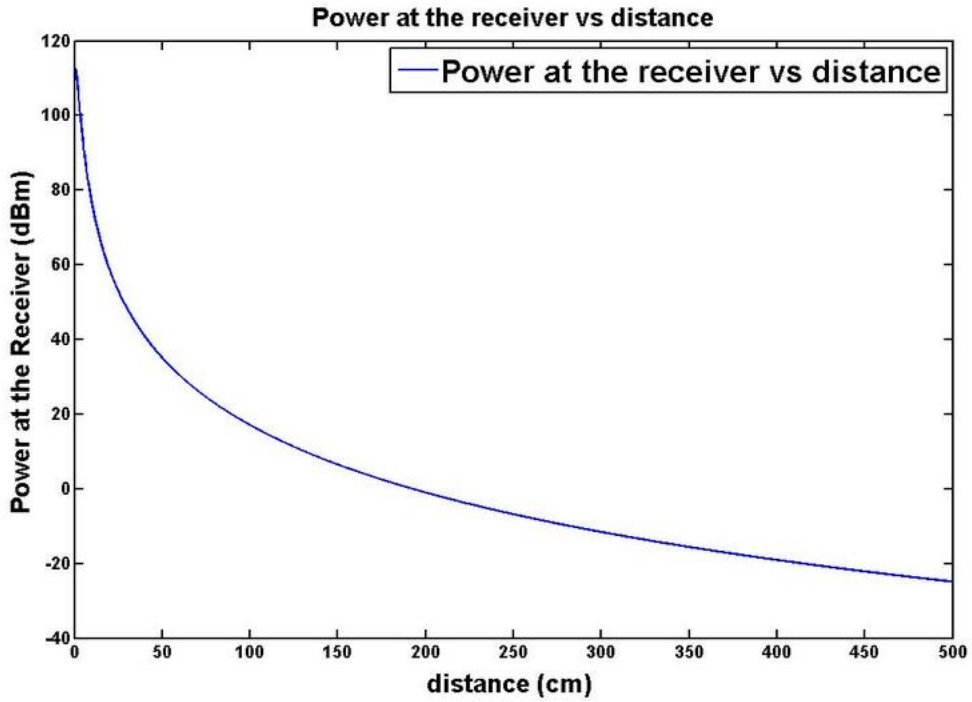


Figure 1: Signal power w.r.t. distance.

that a point source emits the signal uniformly in all directions (isotropic radiator) with a

transmission power P_t. The power density at distance d is equal to the ratio between the transmission power and the surface area of a sphere centered in the source with radius d:

$$F = \frac{P_T}{4\pi d^2} \; [W/m^2]$$

1.7 Antenna gain

An idealized isotropic antenna radiates power equally in all directions, real antennas always have directive effects. The *antenna gain* is the power output, in a particular direction, compared to that produced in any direction by a perfect isotropic antenna. We can talk about **directivity** D and **gain** G as:

$$D = \frac{power\ density\ at\ distance\ d\ in\ direction\ of\ max\ radiation}{mean\ power\ density\ at\ distance\ d}$$

$$G = \frac{power\ density\ at\ distance\ d\ in\ direction\ of\ max\ radiation}{\frac{P_T}{4\pi d^2}} k$$

Where k is the antenna efficiency factor $<= 1$.
Directional antennas point energy in particular direction to achieve:

- Better received signal strength.
- Less interference to other receivers.

1.8 Attenuation w.r.t. distance

Let g_t be the maximum transmission gain. The received power density in the direction of maximum radiation is given by:

$$F = \frac{P_T g_T}{4\pi d^2} \; [W/m^2]$$

$P_t g_t$ is the Effective Isotropically Radiated Power (EIRP) and represents the power at which an isotropic radiator should transmit to reach the same power density of a directional antenna at distance d.
The power received at distance d in case of no obstacles (line of sight) can be expressed with the **Friis equation**:

$$P_r = P_T g_T g_R (\frac{\lambda}{4\pi d})^2 \frac{1}{L}$$

Where P_T is the transmitter radiated power, g_T and g_R the gains of the transmitter and receiver antennas, λ is the wavelength $\frac{c}{f}$ and d the distance between the transmitter and the receiver. Finally, parameter $L > 1$ accounts for hardware losses.
The antenna gain is expressed in decibel (dB). The transmit and receive power in dBm. The path loss is $\frac{transmit\ power}{receive\ power}$ and it's measured in dB.

1.9 Path loss

The formula:

$$PL = (\frac{\lambda}{4\pi d})^{-2}$$

represents the path loss in free space due to geometric spreading. Other attenuations are the result of various types of obstacles and atmosphere absorption (water, vapor, etc...). If we define the path loss as P_R/P_R we can see that the formula above is true if $g_T,\ g_R,\ L = 1$.

1.10 Two ray propagation model

In case signal propagates over line of sight and one ray is reflected, the ratio between received power and transmitted power takes the following form

$$\frac{P_R}{P_T} = g_R g_T (\frac{h_1 h_2}{d^2})^2$$

In the 2 ray model the received power decreases faster than the free space model, $\frac{1}{d^4}$ vs. $\frac{1}{d^2}$.

Figure 2: Two ray propagation.

Anyway, real life propagation is far more complicated, but we can express mean received power with a generalization of the Friis equation, where the propagation coefficient is η instead of 2. The propagation coefficient typically has value between 2 and 5.

$$P_R = P_T g_T g_R (\frac{\lambda}{4\pi})^2 \frac{1}{d^\eta}$$

$$P_r(d)(dB) = 10 log_{10} P_r(d_o) + 10\eta log_{10}(\frac{d_o}{d})$$

1.11 Multipath fading

We already said that the signal while propagating, can encounter different obstacles that affect its wave. Every signal replica received is combined at the receiver and the results depends on:

- The number of replicas.
- Their phases.
- Their amplitudes.
- Frequency.

Different delays from different signal replicas (*delay spread*) can widen the channel impulse response leading to an intersymbol interference. The impact of delay spread can be quantified by computing the root mean square (RMS Delay Spread):

$$\tau_{RMS} = \sqrt{\frac{1}{\sum_{n=1}^{n} P_i} \sum_{n=1}^{n} (\tau_i^2 P_i) - \tau_d^2}$$

With:

$$\tau_d = \frac{\sum_{i=1}^{n} (\tau_i P_i)}{\sum_{i=1}^{n} P_i}$$

Where:

- τ_{RMS} is the RMS delay spread.
- τ_i is the delay on the path i.
- P_i is the power received on path i.
- n is the number of paths.

The *coherence bandwidth* is a statistical measurement of the bandwidth interval over which the channel is *flat*. If this measure is bigger than the signal bandwidth the channel is flat, otherwise we have delay spread and thus intersymbol interference and reception errors. To ISI we introduce equalization that raises up the complexity.

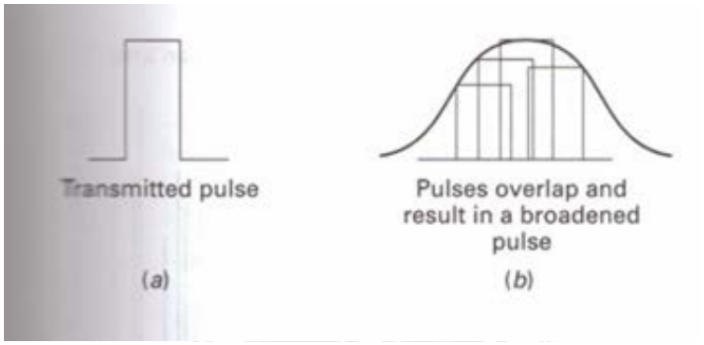


Figure 3: Broadened pulse.

2 Techniques for energy efficient communications

2.1 Energy efficient protocols

Despite the improvement in battery tech, the problem of the power supply has not been solved and cannot be solved only on the battery side. Energy demand is increasing and also the user expectations in terms of device lifetime.
With the term *network lifetime* we refer both to the time until the first node in the network dies due to battery drain; and the time before the network disconnects or fails to perform critical tasks.
When we talk about energy efficiency we want to express how good a given task is performed, thus we can look at the energy spent by the network to deliver a bit to the final destination. Energy efficiency is to be considered in combination wit other metrics, such as throughput and latency.
Network related energy consumption has two components:

- **Computing:** in network data processing, data fusion and aggregation, protocol operations.
- **Communications:** wireless transceiver consumes energy either to transmit and receive data and control packets, or when it is idle, ready to receive.

Thus there is a the need to find the good trade-off between computation and communication. Energy-efficient communication protocols can add overhead and computational complexity and also the placement of the computing is an issue. The data can be processed in network (higher energy consumption due to computing, lower energy consumption due to compact data transmission) or in the devices.
We can significantly decrease overall energy consumption in case of long range communication

by applying power control.
Wireless technologies can dynamically change the modulation scheme used over time. Use of high data rate modulations reduce the time needed to transmit packets and it's thus associated with energy consumption.
Also HW-dependent optimization or even the selection of the HW itself can affect the performance in terms of energy consumption.
Several protocols for ad hoc network routing exploit the idea of operating wireless interface card in promiscuous mode, i.e. passing received packets to higher layers and process them even if they are not addresses to the node. This is done in order to gather information about the best broadcast channel to use to optimize operations. Doing this forces the interface card to stay in idle, instead of lower power modes, for long periods of time, leading to significant energy consumption.
A wireless transceiver should instead stay in a low power *sleep state*, a mode where it cannot receive or transmit packets whenever information exchanged in a handshake make it aware that it's not involved. This is also why the destination address is the first field of the header.

2.2 MAC

There is an awake/asleep schedule. The awake mode involves transmit, receive and idle. It is an high energy consuming state in which the transceiver is ON. The asleep mode is a state in which the transceiver is OFF and can't receive nor transmit packets. It is a low energy consuming state.
Thus we can describe a *duty cycle* as $\frac{T_{ON}}{T_{ON}+T_{OFF}}$.

We have two possible classes of protocols:

- **Synchronous:**
  - Nodes exchange information to coordinate on when to wake up.
  - A periodic control message exchange ensures that nodes will wake up.
  - A packet is transmitted to a neighbour when it is ON.
- **Asynchronous:**
  - Awake/asleep schedule of neighbours is unknown.
  - No control is needed to keep information updated (removed one overhead).
  - To ensure reliable communications we must send packets until the destination node wakes up (added an overhead when sending). Otherwise nodes must follow an approach that consists in selecting one neighbours between the awaken ones.

Nodes not involved in communication should go to sleep. Nodes should minimize collisions. There is a compression of the header in order to keep the transceiver ON as little as possible. There is a limit on information exchanged that leads to the aggregation of redundant data.
MAC tends to increase latency. The reception state is more energy consuming than the transmission, because it's not possible to predict when a packet has to be received.
Transceiver can be in one of the following states:

- **tx:** awake and transmitting.
- **rx:** awake and receiving.
- **idle:** awake neither transmitting nor receiving.
- **asleep:** the transceiver is not operational. There can be several asleep states with different parts of the circuitry switched OFF.

2.3 Energy efficient techniques

2.3.1 Data Link

If a channel is in a bad state (deep fade) it is convenient to delay transmission as it is very unlikely that packets will be correctly received. Energy efficient ARQ and FEC schemes optimize energy consumption ensuring reliable communication at the same time.

2.3.2 Routing

It can be more energy efficient to transmit over higher number of shorter links or minimize the number of hops. It depends on the scenario. Routing can minimize the overhead associated to route discovery and contributes to balance the load among nodes to increase network lifetime. Energy aware routing solutions can consider residual energy to select the best next hop relay and link quality aware solutions can also avoid re-transmissions. The selection of relays has to help data fusion or aggregation. If we can combine all of these aspects we have a *cross-layer* solution.

2.4 TailEnder

Combined use of 3G and WiFi, predicting WiFi availability.

3 Cellular systems

3.1 GSM

Second generation (2G) cellular system. Eight channels per carrier, each one with 200KHz bandwidth. Every channel is split in time-slots assigned to 8 different users (i.e. TDMA and FDMA). Supports power control, i.e. the possibility to adjust the power emitted from stations according to the conditions of propagations. Uses **discontinuous transmission** in order to save power (during pauses in speech, voice transmission is interrupted).
Frequency Division Duplexing (FDD): lower part of frequency band is used for **uplink** (to base station) and the higher part for **downlink** (from base station). This because transmission at higher frequency consumes more and thus it's better to reserve that for base station.
Each TDM frame is 4.615 ms and hence each slot is 577 μs long, mainly meant to voice transmission.

3.2 Architecture

The network contains:

- **User Equipment** (UE) or Mobile Station (MS): interfaces the user and handles radio functionality.
- **Access Network** (AN): communication to and from the user equipment, handles all radio functionality.
- **Core Network** (CN): communication between AN and other networks, handles switching and routing.

Services and applications are above the network.

3.2.1 Areas

- **PLMN - Public Land Mobile Network Area**: service area of a cellular network.
- **MSC/VLR Area**: area managed by an MSC. Users data are temporarily stored in a database called VLR.
- **Location Area**: a MSC/VLR area is divided in more Location Areas. If a user changes LA it has to perform a location update. LA are identified by a LAI (LA Identifier), transmitted by the BTS of the LA over the broadcast control channel.
- **Cell**: area covered by a BTS. Identified by a BSIC (Base Station Identity Code), transmitted by the BTS over the broadcast control channel.

3.2.2 Mobile Station - MS

It's the terminal owned by the user. Three categories depending on nominal power:

- *Vehicular*: antenna that can emit up to 20W.
- *Laptops*: the antenna can emit up to 8W, transportable but power demanding to operate.
- *Personal*: the antenna can transmit up to 2W, such as mobile phones.

Multi Band MS can operate on different frequency bands and multi slot MS over different channels, in different slots. MS is composed of an ME (Mobile Equipment) and a SIM (Subscriber Identity Module), where:

- ME is the terminal through which a user gets access to the network. It is identified by the IMEI (International Mobile Equipment Identifier).
- SIM activates the terminal for a given user and stores all the needed information. It is a smart card with a processor and a memory, it has different formats. It stores:
  - *Serial number*: uniquely identifies SIM card.
  - *International Mobile Subscriber Identity - IMSI*: uniquely identifies the user in the network. It has 3 fields:
    - MCC - Mobile Country Code: 3 digits.
    - MNC - Mobile Network Code: identifies the operator (2 digits).
    - MSIC - Mobile Subscriber Identification Number: identifies the SIM (up to 10 digits).
  - Security authentication and ciphering information: A3 and A8 algorithms to perform authentication and encryption. K_i, K_c, keys for authentication and encryption.
  - Temporary network information:
    - LAI (Location Area Identifier): last visited location area id.
    - TMSI (Temporary Mobile Subscriber Identity): temporary id assigned by the network.
  - List of services to which the user subscribed.
  - PIN and PUK.
  - Access rights.
  - Prohibited networks.
  - Call messages.
  - Phone numbers.

  A ME without a SIM is enabled to make only emergency calls.

3.2.3 Base Station System - BSS

It includes the units involved in radio coverage and communication. It also performs radio resource management. It includes:

- Base Transceiver Station - BTS. HW/SW components for transmission and reception through the interface. It has only executive tasks. It implements the low-level protocols of the radio interface. It performs frequency hopping and encryption. It also performs quality measurements on the physical channels reporting them to the BSC, that makes the decisions. It can broadcast on a control channel the System Information message, that contains required data for the MS to access the network. It is also in charge of sending paging messages to locate the position of the user. It interfaces to the BSC via PCM channels at 64 kbit/s. It connects the PCM channels with the ones of the radio interface.
  It is usually divided into:
  - *TRX* (transceiver): radio elements responsible for reception and transmission on a single radio carrier:
    * Transmitter: modulation, power amplifier, ...
    * Receiver: diversity, demodulation, ...
    * Signal processing.
    * TRX controller.
    * *BCF* (Base Common Functions): they controls element of TRX for synchronization and frequency hooping computation. They also interface with the BSC.
- Base Station Controller - BSC. It monitors and manages the resources of a group of BTSs, from which it receives info about the state of the interface. It is responsible for reservation/release of radio channels, handover, transcoding, ...
  One BSC controls a large number of BTSs (~10-100)- Its main tasks are:
  - The configuration of each cell by assigning traffic and control channels.
  - The set up and release of connection between channels.
  - The management of handovers between BTSs.
  - The management of the paging messages.
  - The analysis of the link quality and power level measurements performed by the BTSs and MSs.

  The BSC is concerned with the management of radio resources. From a functional point of view it is a switching node, but doesn't perform routing calls (that is instead job of the MSC). Instead, it connects the circuits of the BTS with those of the MSC, possibly carrying out the transcoding. It switches the circuit in case of handover (intra BSC).
  The BSC can be placed where is an MSC, near BTSs or as a standalone.

3.2.4 Transcoder Rate Adaption Unit - TRAU

The GSM voice coding is 13 kbps while the PCM is 64 kbps. The transcoding is performed by the TRAU, that most likely will be in the BSC, but can also be in the BTS. In this case, the 13 kbps flow must be transported in the channels at 64 kbps. On each PCM channel, four 13 kbps flows are multiplexed after being transformed in 16 kbps streams to add redundancy. For each GSM carrier (8 channels, 13 kbps) we need 3 PCM channels at 64 kbps:

- One for the signal carried by control protocol LAPD.
- Two to carry the information of the 8 multiplexed channels.

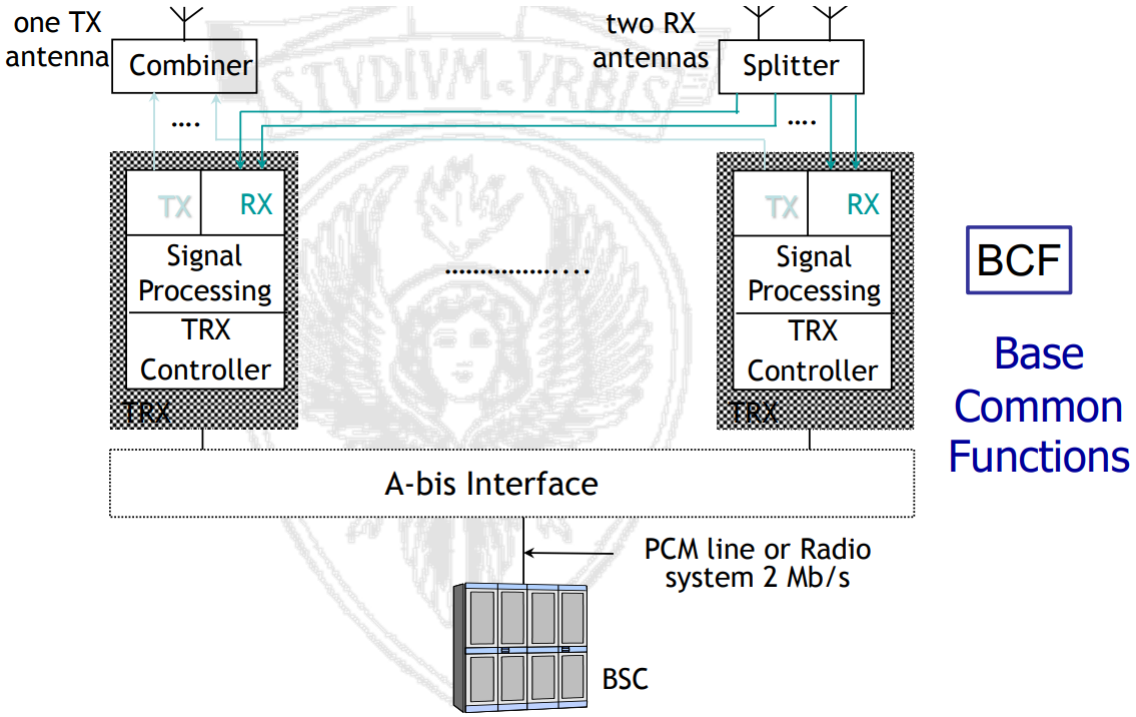


Figure 4: BTS functional scheme.

3.2.5 Network Switching Subsystem - NSS

It is the subsystem responsible for circuit switching to the mobile users. It includes:

- Mobile Switching Center (MSC): telephone switching center for mobile users. It is a switching element that performs mobility management. It is normally associated with a VLR. The MSC is connected to the BSC of its area and to other MSCs. The connection uses PCM channels. One or more MSC for each PLMN networks is interfaced to the fixed telephone network for routing. An MS can be reached by fixed users using the phone number (MSISDN). The call is routed to the GMSC, that identifies the HLR that has the user information associated with the MSISDN. The HLR returns the MSRN (Mobile Station Roaming Number) that has the VLR/MSC associated to the user. This is queried to get the MSRN that is assigned by the VLR. The MSRN allows the GMSC to route the call to the MSC area where the user is located.
  The MSC provides:
  - *CM (Connection Management)*: originating call, terminating call, gateway.
  - *MM (Mobility Management)*: location updating, periodic registration, authentication.

  The MSC implements protocols to exchange info with other elements of the network: DTAP (exchange with MS), BSSMAP (exchange with the BSC), MAP (exchange with other elements).
- Visitor Location Register (VLR): a database that contains info about users in the area managed by the MSC. It is a temporary database that contains data to serve the MS currently under the MSC associated with the VLR. All user under that MSC/VLR area are replicated in the VLR, mapping the IMSI in a TMSI to protect from intrusions. The TMSI is changed every time the location is updated.
- Home Location Register (HLR): the main database responsible for storing the information of mobile users. It contains the info needed to identify the VLR. It is permanent and

uniquely associated to a GMSC. It stores information for all MSs that are at that GMSC. It stores the IMSI, SIM id, etc... Its main tasks are:

  - Managing localization storing the VLR number of users.
  - Sending routing information (MSRN) to the GMSC.
  - Registration, cancellation and activation/deactivation of additional services.
  - Storage of authentication and encryption parameters.

- Authentication Center (AuC): usually associated with the HLR which contains the keys and the procedures to authenticate a mobile user. The AuC computes the keys for authentication and encryption. Stores the secret keys K_i for each user, generates random numbers and calculates $SRES$ and the encryption key K_c. It provides the triplet to other network elements.

- Equipment Identity Register (EIR): contains the IMEI of alla devices authorized to access the service. It's a database whose use is at discretion of the operator. It contains the id and characteristics of GSM terminal equipment. It can be used to protect the network from not compliant uses. It keeps:

  - **White list**: valid IMEIs with corresponding MEs that can be used in the GSM network.
  - **Black list**: IMEIs of alla MEs that can't use the GSM network, with the exception of emergency calls.
  - **Gray list**: IMEIs that correspond to MEs that can be used, but that, for some reason need to be tracked.

3.2.6 Security procedures

Authentication It is the task to verify user's identity and protect against fraudulent use of identification.

Encryption The confidentiality of the communication is ensured applying encryption algorithms and frequency hopping.

The user anonymity is ensured also through the use of temporary id numbers.

- K_i: user authentication key of 128 bits stored in the SIM and AuC.
- $RAND$: 128-bit random number generated by the AuC and then sent to the MSC.
- $A3$: authentication algorithm stored in the SIM and AuC. It ouputs the SRES.
- $A8$: algorithm that determines the encryption key K_c which is stored in the SIM and AuC.

The $(RAND, SRES, K_c)$ triplets are generated for each IMSI and stored in the HLR.

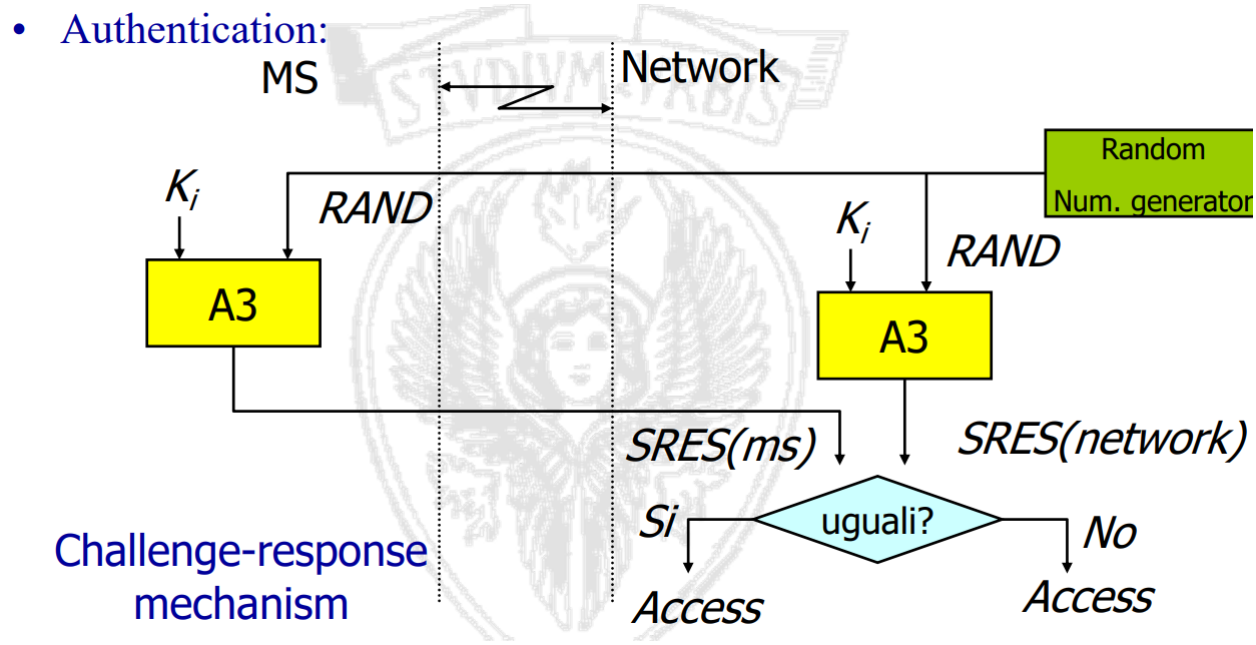


Figure 5: Authentication.

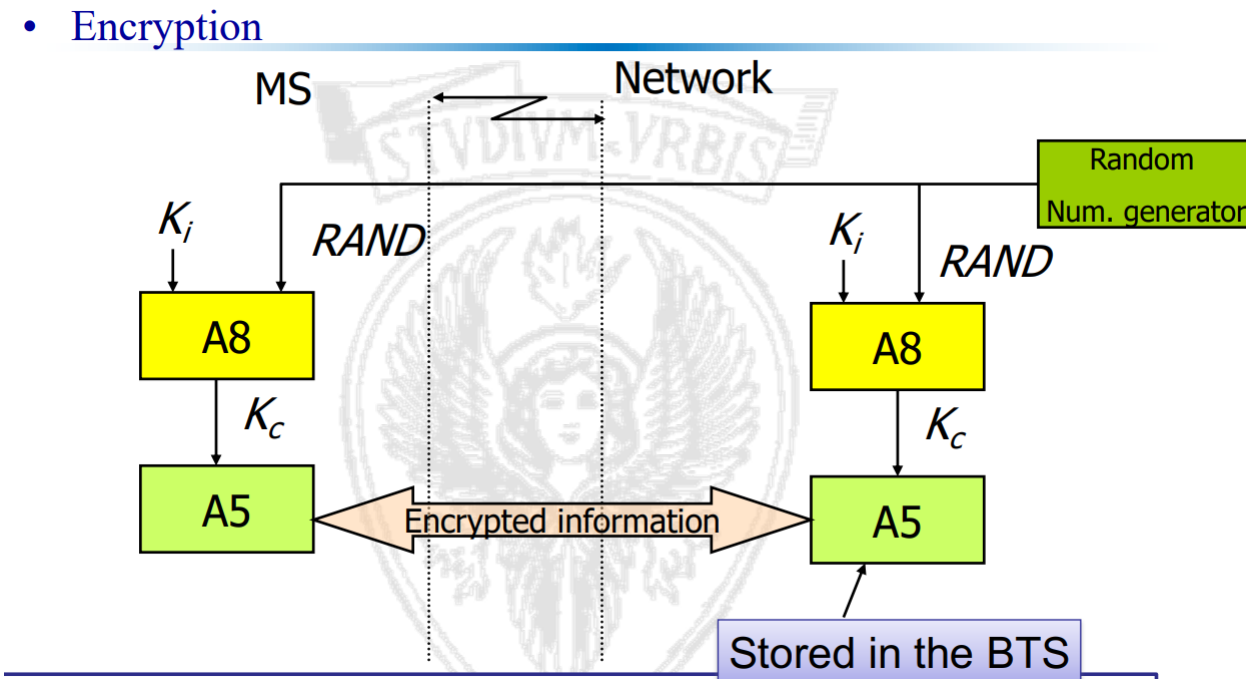


Figure 6: Encryption.

3.2.7 Operation and Maintenance Subsystem - OMSS

Includes the functional units responsible for monitoring the network. It has network measurement and control functions and it's monitored and initiated by the OMC (Operation and Maintenance Center). Basic functions:

- *Network administration*: configuration, operation, performance management, statistics analysis, maintenance.
- *Commercial operation and charging*: accounting and billing.
- *Security Management*: EIR management.

4 Logical channels

Uniquely identify the type of information they carry:

- Signaling.
- Data traffic.

Channels types:

- Traffic channels vs. control channels.
- Common channels vs. dedicated channels.

4.1 Control channels - CCH

Control channels carry signaling information, there are 14 different types. We can divide them in 3 main categories:

- **Broadcast Channels - BCH**: unidirectional downlink channels that provide general info about the network. We can further identify them:
  - *Frequency Correction Channel - FCCH*: downlink channels used to correct MS frequency, 148 bits w/o coding.
  - *Synchronization Channel - SCH*: carry the BSIC (Base Station Identity Code) and the frame number (FN), 25 bits + channel coding.
  - *Broadcast Control Channel - BCCH*: carry general info broadcasted to all user of a BS. 184 bytes after coding.
- **Common Control Channels - CCCH**: carry info to start a connection. We can further identify them:
  - *Paging Channel - PCH*: downlink channel used by the BTS to notify an incoming call to a MS, broadcasted over a LA.
  - *Random Access Channel - RACH*: uplink channel used by a MS to request access to the network. Prone to collisions. The access is random, thus is not coordinated with other MSs. Messages correctly received by the BS are acked on the AGCH channel. RACH messages include a temporary random sequence that is included on the ACK sent on the AGCH channel. The transmissions use the slotted ALOHA protocol.
  - *Access Grant Channel - AGCH*: downlink channel carrying reply to RACH requests.

- **Dedicated Control Channels - DCCH**: carry signaling info specific of a single connection. We can further identify them:
  - *Slow Associated Control Channel - SACCH*: bidirectional channel used to exchange connection metrics between MS and BS.
    - Downlink: power control commands and BCCH information. We send these types of messages:
      - Power command.
      - Time advancement.
      - Frequency hopping sequence.
      - Frequencies used by adjacent channel.
    - Uplink: MS measurements report:
      - RXLEV-SERVING-CELL.
      - RXQUAL-SERVING-CELL.
      - RXLEV-NCELL "N".
      - BCCH-FREQ-NCELL "N".
      - BSIC-NCELL "N".
      - Frame Error Rate (FER).
      - Received signal level from neighbour cells.
  - *Fast Associated Control Channel - FACCH*: used for exchange time critical info, this channel steals capacity from the associated traffic channel.
  - *Stand-alone Dedicated Channel - SDCCH* : control channel assigned after a RACH request.
- **Traffic Channels - TCH**: carry speech and data. We have two different types:
  - *Full Rate Channels*: gross rate of $22.8Kb/s$.
  - *Half Rate Channels*: gross rate of $11.4Kb/s$.

4.2 Channel coding: voice channel at $13Kb/s$

The bits are not contiguous output of the coding process, but they are interleaved to avoid the loss of whole messages.

4.3 From logical to physical channels

Signaling requires lower bit rates than user transmissions. The actual transmission rate may be reduced by using *multiframes*. The slots are associated with IDs and may be assigned over a period of multiple frames, i.e., over a multiframe.

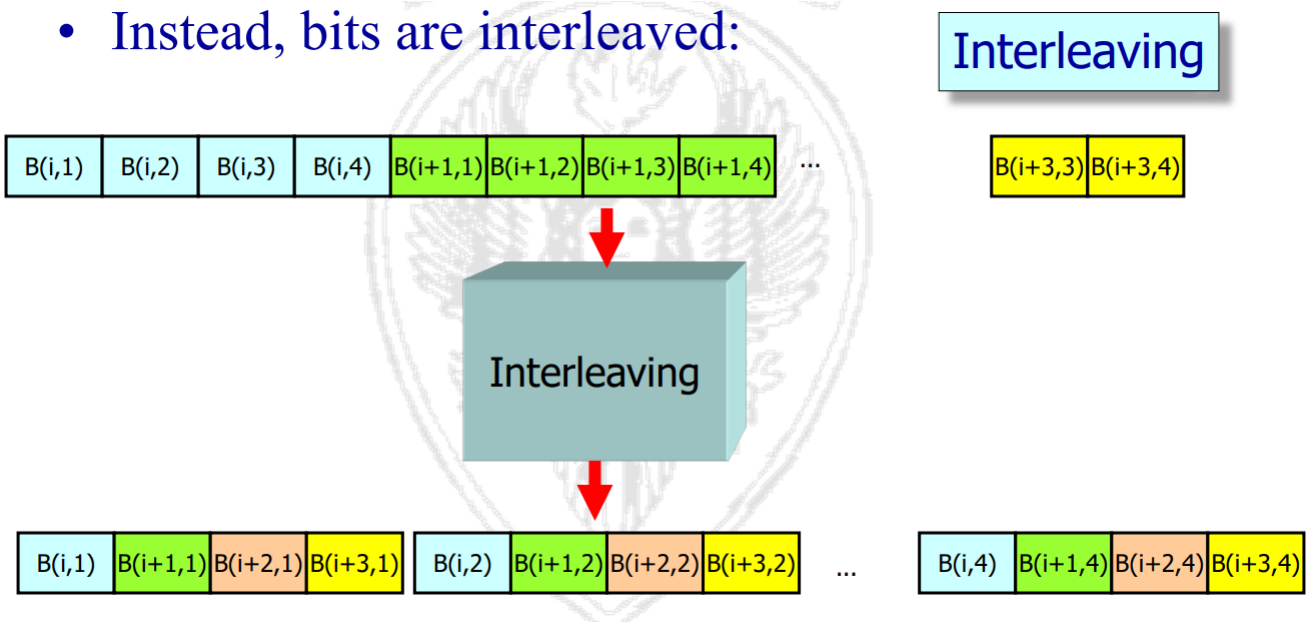


Figure 7: Interleaving.

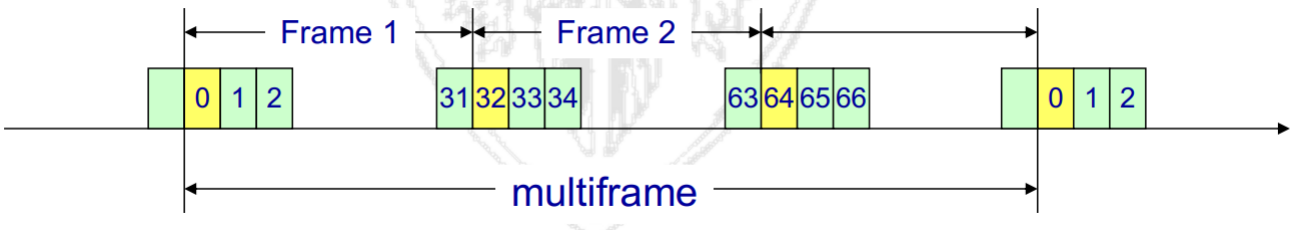


Figure 8: Multiframe.

4.4 Multiframe example: SACCH

A *data burst* (name of packets in GSM) carries 114 bits of data.

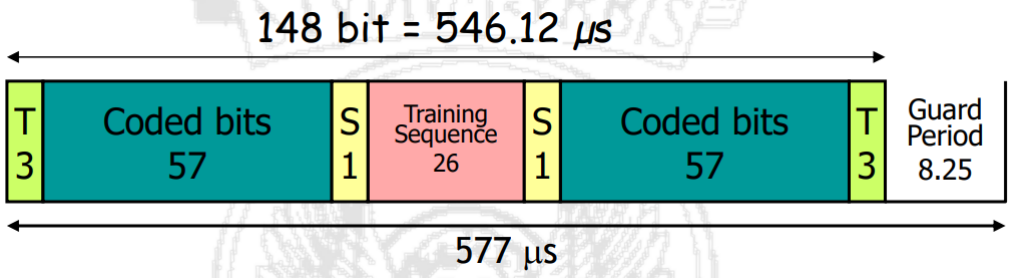


Figure 9: Data burst.

- A channel using one slot per frame has a rate of 114 [bit]/4.6 [ms]=24.7 Kb/s
- Coded speech is transmitted at a rate of 22,8 Kb/s.
- 1,9 Kb/s are not used, equal to 1 SLOT every 13 frames
- SACCH: 1 SLOT every 26 frames = rate of 950 bit/sec.

An example of multiframe TCH full duplex:

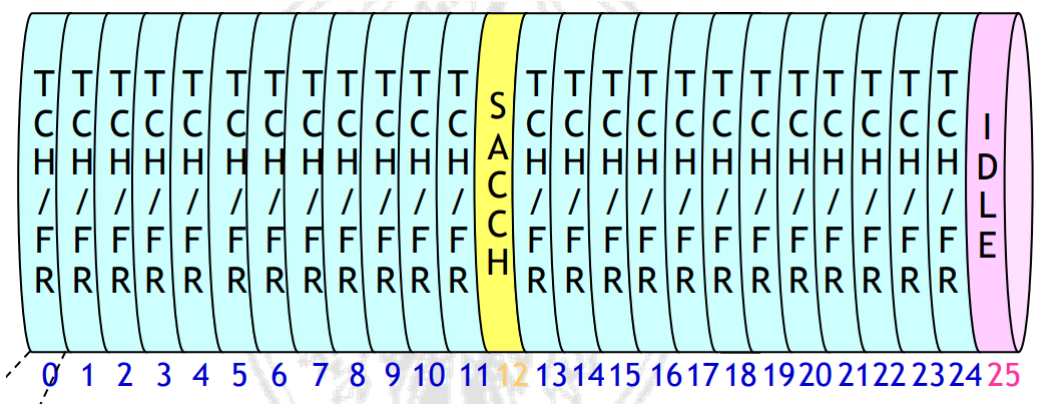


Figure 10: Multiframe full duplex.

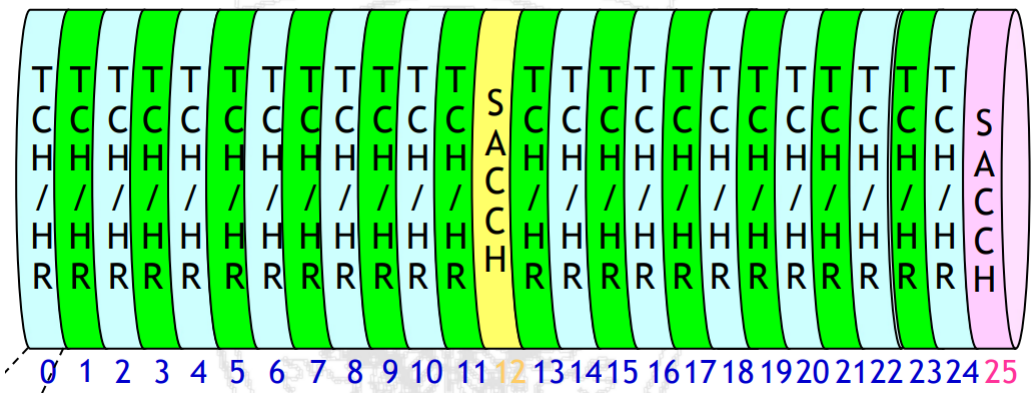


Figure 11: Multiframe half rate encoding.

4.5 Common signaling channels

Slot 0 over the C0 carrier (or main carrier) is used to obtain one or multiple channels that use a multiframe with 51 frames. The main carrier is always transmitted with the highest power among carriers, which allows a MS to synchronize with it. Another slot is used to obtain 8 SDCCH that are used for setup and other messages.

4.6 Physical blocks (bursts)

The physical block is the info transmitted during a slot. Due to TDMA, each block is an autonomous entity that has to be transmitted at an appropriate power level to avoid interference. We have various types of bursts:

- **Normal Burst**: used for transmissions over traffic channels. The tail bits (T) are always

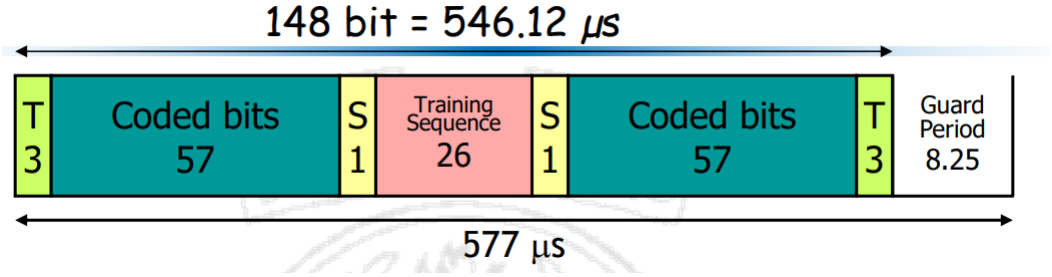


Figure 12: Normal Burst.

set to 0. The stealing bits (S) indicate if the burst contains user data or signaling info. The coded bits are for user and are 13 kbit/s for speech and 9.6 kbit/s or lower for data. The training sequence are control bits used for equalization and tuning of the transmitters.

- **Access Burst**: used to transmit info over a RACH and for first time access. The access to it is asynchronous. It contains 156.25 bits.

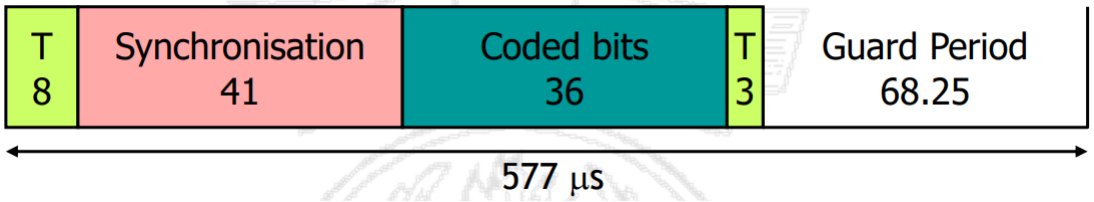


Figure 13: Access Burst.

- **Frequency Correction Burst**: it is used over the FCCH, 142 bits set to 0. Corrects the frequency of the MS's local oscillator, locking it to the one of the BTS.

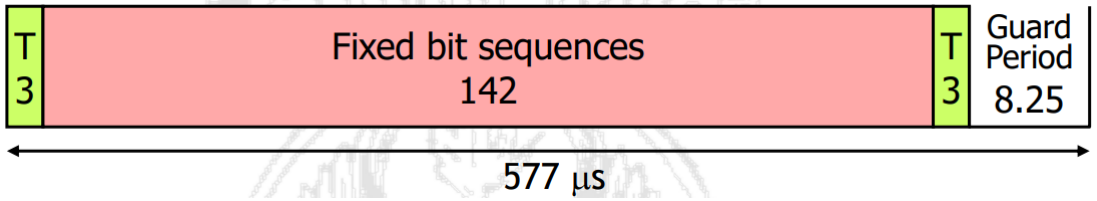


Figure 14: Frequency Correction Burst.

- **Synchronization Burst**: it is used to transmit info about synchronization for slots and frames.

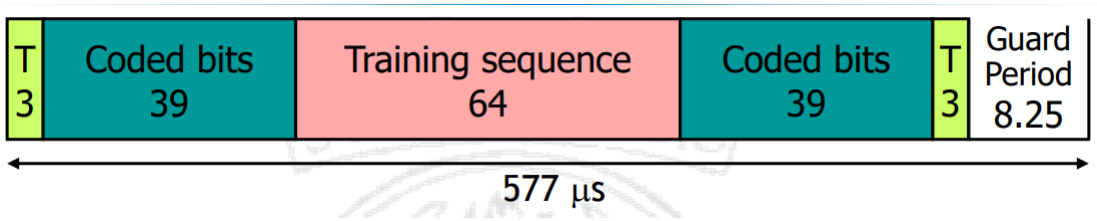


Figure 15: Synchronization Burst.

- **Dummy Burst**: it contains no information, only padding bits. It is used when there's no data to be carried and there are unused timeslots of the BCCH carrier.

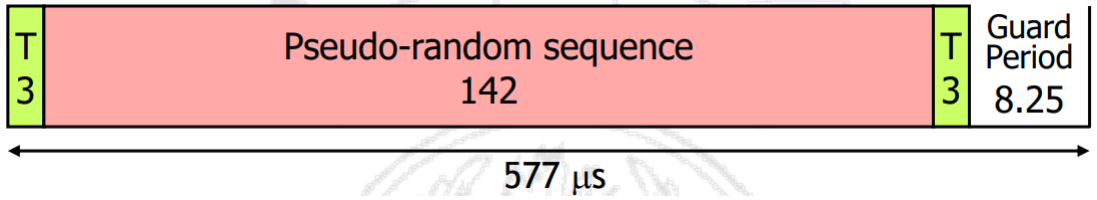


Figure 16: Dummy Burst.

5 Radio Interface

5.1 Frequency Hopping

Multipath fading depends on the carrier used for transmission. When transmitting to a user, carriers can face high or low attenuation. Since we use FEC codes to increase the transmission robustness, it's better to have errors (due to high attenuation) spread over multiple flows of info, similarly to the reasons behind interleaving. Frequency hopping changes the carrier used for transmission on a per slot basis, according to a random sequence.

5.2 Power Control

The output power of the MS is controlled by the BTS, that sends power control commands to it. The increment/decrement step is 2 dB. The goal of power control is to bring power received by the BTS to a predetermined level, this reduces interference and energy consumption.

5.3 GSM Synchronization

It divides in multiple parts:

- **Carrier Frequency Synchronization**: each MS must retrieve the frequency of the carrier. The frequency of the radio carrier is obtained by the MS listening to the BCCCH. On this channel a special sequence of bits is transmitted at high power to select the carrier frequency and then adjust the frequency of the local oscillator.
- **Slot Synchronization**: each MS must have info on the current slot. Up/down link transmissions have propagation delta (τ) due to the distance between BTS and MS. Each

slot needs a guard period to compensate for synchronization errors. We can make a conservative selection setting the guard time to $T_g = \max_i(2\tau_i)$. The GSM network is designed to have cells with $R_{max} = 35Km$. In the worst situation there is a guard time of $2\tau = 2 \times 35/3 \times 10^8 = 233\mu s$ which corresponds to 68.25 bits at the rate of 270.8 kb/s.

  - *Timing advance*: to limit guard time the BTS estimates the delay and sends the info to the MS which can compensate by anticipating the transmission. The technique is used in GSM is called *timing advance* in which as the MS moves away from BTS, it anticipate the transmission.

- **Frame Synchronization**: each MS must know the frame number. For both frame and slot synchronization we have to cope with multiframe architecture. The sequence of frequency hopping depends on this. The MS has the need to know the number of the current frame, thus the BTS transmits the info needed to the MS to let it know the slot and the FN.

- **Base Station Synchronization** (optional): the BSs have synchronous clocks and same frame number.

6 Procedures

6.1 IMSI attach

When a MS is switched ON:

- **Cell selection**: the MS selects the BTS to which it wants to tune.
  - The MS scans all RF carriers operating in the cell and thus scans c0 carrier that is where BCCH is transmitted. Such carriers are transmitted at higher power and frequency hopping is disabled.
  - The MS connects to the RF Carrier from which it receives the strongest signal.
  - Through the FCCH channel the MS synchronizes to the BTS carrier.
  - Through the SCH the MS synchronizes to the slot and frame and receives the BSIC (Base Station Identity Code).
  - The MS can now decode the BCCH, which includes LAC (Location Area Code), CGI (Cell Global Identity), MCC (Mobile Country Code), MNC (Mobile Network Code).

- **Registration** (Location Update Procedure): the MS notifies the MSC of its presence in that LA. We have two cases that depend on the received LAI:
  - If it is the same of the one stored in the SIM (we turn off and on the phone in the same LA), then the IMSI attach procedure is invoked and the MS activates its IMSI stored in the VLR.
  - If there's no LAI stored or we receive a different one from the one we store we invoke the *Location Update* procedure.

6.2 Location Update

It is performed when a MS is switched on (and 2nd case as above) and periodically. If it's not successful, the VLR flags the user as detached. We have 2 different types:

- Two LAs of the same MSC/VLR (simplest):
  The System Information Message sent over the BCCH contains the LAI. Once tuned to

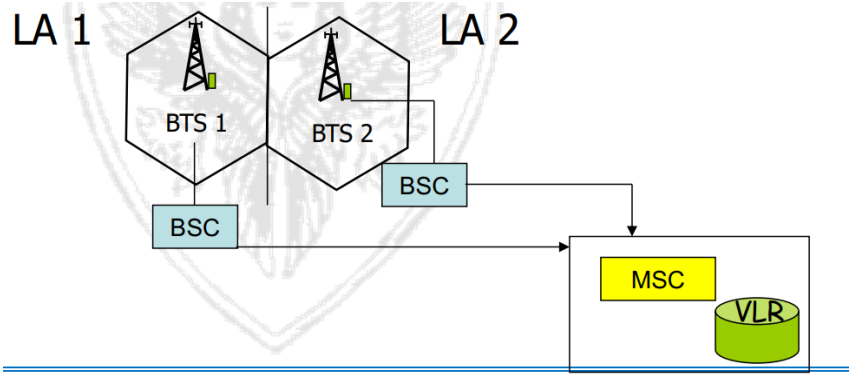


Figure 17: Same MSC/VLR.

  a new BTS, the MS can determine if a location update is needed.

- Two LAs of different MSC/VLRs:

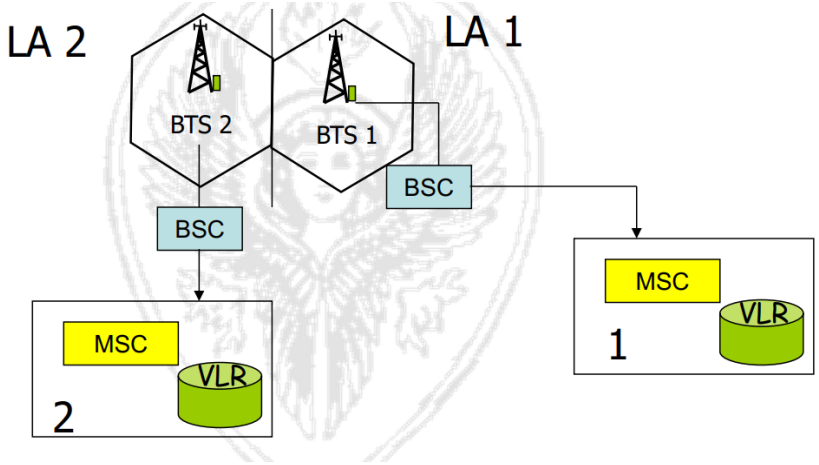


Figure 18: Different MSC/VLR.

6.3 Call set up

6.3.1 PSTN-originated Call

1. The PSTN/ISDN user dials the Mobile Subscriber International ISDN Numbers of the user it wants to call.

2. The dialed number is analysed by the PSTN/ISDN network, which routes the call to the GMSC oof the PLMN of the called user, by making use of the National Destination Code (NDC).

3. The GMSC receives the message requesting to setup a call through the SS7 network, which contains the MSISDN of the user called.

4. The GMSC identifies the HLR containing the data of the called user.
5. The GMSC sends `send routing info` message to the HLR.
6. The HLR identifies the address of the VLR in which the called MS is registered.
7. The HLR sends a `provide roaming number` message to the MSC/VLR.
8. The MSC/VLR temporarily allocates a Mobile Station Roaming Number (MSRN) to be used for the call.
9. The MSRN is forwarded by the MSC to the HLR.
10. The GMSC routes the call towards the MSC/VLR of the LA in which the MS is currently located.
11. The MSC/VLR activates the paging procedure thanks to which it identifies the currently visited LA thanks to the IMSI and sends a paging command to all the BSCs of the LA.
12. BSC requires the BTSs to send the paging message destined to the MS over the paging channel (PCH). The message contains the TMSI assigned to the MS.
13. The MS replies to this message by requiring a SDCCH through the RACH.
14. The MSC/VLR activates the authentication and the ciphering procedures.
15. A traffic channel (TCH) is allocated for communication.
16. The MSC/VLR notifies the caller that the called phone is ringing.
17. The called user answers the call.
18. The connection between the 2 users is established.

6.3.2 MS-originated Call

1. The called number is dialed by the MS.
2. The current MSC analyses the caller data and authorizes or denies the call. Depends on this, it starts the call routing procedure.
3. If the called number is in the same GSM network, a `send routing info` procedure is started to obtain the MSRN (same procedure as PSTN calls).
4. If the called number is in another GSM network, the call is routed to the GMSC.

7 Ad Hoc Network

It is a wireless multi-hop network without an infrastructure. Devices connected to the network act both as source and as destination and as relays for packets. Those, if on a $s \rightarrow z$ route, are generated by a node s and addressed to a node z.
The pro of this network is that we don't need an infrastructure, thus we have low costs and enables communication where it's usually not viable. Besides, they are also very dynamic, hence they can deal with mobile devices and implement energy saving modes. They don't need high energy, have a low overhead and use simple protocols.
One major issue is scalability because these networks can grow up to thousands of nodes. The code must be simple, almost inexpensive, also due to small storage capability.

An ad hoc network must be: self-organizing, self-configuring and self-maintaining.
It can be applied in the following scenarios:

- Disaster recovery applications
- Military networks
- Personal Area Networks
- Home Networking
- Wireless Sensor Networks (WSNs) and IoT
- Inter-vehicular communication
- Mesh Networks (extension of WiFi standard)

7.1 MAC

The approach is CSMA/CA. We don't use CSMA/CD because wireless network can't detect collisions during transmission. Also a TDMA approach is not the best, it would require a synchronized environment; it's not a good idea: nodes can be a lot, can move, and the overhead would significantly increase.

7.1.1 Distributed Coordination Function

Before transmitting a frame, the sender node X performs carrier sensing. If the channel is free for a certain amount of time (Distributed InterFrame Space - DIFS), X will transmit the packet. Otherwise, X will wait for the end of the current transmission, plus a random back-off time. The back-off timer is frozen if X senses the channel busy and when it goes to 0, X can transmit. The timer value is picked randomly within a window interval of CW slots. At the first attempt CW is set to 16 slots (minimum) and we use an exponential back-off. At every other transmission attempt CW is doubled until it reaches 1024 slots (maximum).
X can understand if the packet has received thanks to an ACK sended after a SIFS (Short InterFrame Space) time.

7.1.2 Virtual Carrier Sensing

To mitigate performance impairments due to hidden terminal issue, DCF (that is WiFi CSMA/CA) uses virtual carrier sensing. Before transmitting, the node performs carrier sensing, waiting for DIFS time. Then it transmits a short control packet called Request To Send (RTS), informing its neighbours that it will transmit a packet to the destination.
RTS includes a Network Allocation Vector (NAV) field whose value expresses the time from the RTS reception to the ACK reception. By receiving the NAV, all nodes can estimate when not to transmit to not interfere with the ongoing transmission.
If the receiver receives correctly an RTS, it waits for SIFS time and then transmits a Clear To Send (CTS) message. All destination neighbours will know that the channel is busy and for how long. CTS includes a NAV field.
When the sender receives a CTS, it transmits the DATA packet after SIFS. When the destination receives the DATA packet, it waits for a SIFS and then sends the ACK. If the handshake is not completed, the sender performs a re-transmission after a backoff.

7.2 Routing

Intra-AS routing (AS = Autonomous System). We have 2 approaches:

- **Link State approaches** : each switching node (i.e. every node in an ad hoc network) keeps information on the topology and then applies shortest path algorithms (e.g. Dijkstra) to construct its own routing table. Each node periodically sends information on its neighbors (and the associated cost on the links to them) to every node in the network (via flooding). They also send updates if there are some changes. When each node has the complete view of the network topology, it can locally compute the best route with Dijkstra. Then the node fills the routing table accordingly.
- **Distance Vector approaches**: switching nodes exchange information (such as the routing table and hop count) and choose the best route based only on the distance; this approach uses the Bellman-Ford algorithm to determinate the best route. Using only the length of the path, without further context, makes this approach not a good idea in networks where the topology is dynamic.

7.2.1 Count to Infinity Problem

In the second approach, the use of the length of the path alone, can lead to a big problem that is called *count to infinity*. To explain this, let's put in a possible situation. We have 3 nodes x, y, z where the path costs ($Cost(i, j)$) are as follows:

- $Cost(x, y) = 1$.
- $Cost(y, z) = 1$.
- $Cost(x, z) = 2$.

The x and y nodes exchange info as follows:

- x says "the shortest path from me to z costs 2."
- y says "the shortest path from me to z costs 1."

Based on this, if we want to route a packet from x to z we start on x, then go through y and eventually reach z. Now, due to a problem, the path between y and z is somehow blocked. The y node understands that it has to find a different path from itself and z. Looks up in the routing table and sees that from x to z the path costs 2, because it doesn't know that path passes through itself. y can now communicate the new cost to x saying "I found a path from me to z that costs 3 ($y \rightarrow x \rightarrow z$)". x now updates its info and calculates the new cost, finding out that it can now reach z spending 4 ($x \rightarrow y \rightarrow x \rightarrow z$). y sees the new cost and re-updates its info, this will go on endlessly.
We can find some solutions, such as:

- *Bounded network diameter (RIP)*: it is possible to use a TTL and discard packets that went through more than n hops. If the network diameter is limited, the convergence in case of count to infinity is fast.
- *Split horizon with poison reverse*: limits to transmitted info. If A uses info received from B to select route towards D, A won't communicate valid route lengths to B or it will communicate infinity. This doesn't solve all loop situations and disables the broadcast to send updates.
- *Trigger updates* instead of sending periodic updates, they are transmitted immediately in case route length change.

7.3 Goals of Routing in Ad Hoc Networks

- Multi-hop path routing capability.
- Dynamic topology maintenance
- "No loops"
- Minimal control overhead
- Low processing overhead
- • Self-starting

7.4 Proactive Approach

It is based on traditional distance vector and link state protocols. Each node maintains route to each other node in the network. There are periodic or event-triggered routing updates. There is an higher overhead and a longer route convergence time.

7.4.1 Highly Dynamic Destination-Sequenced Distance Vector (DSDV) Routing

The difference with Bellman Ford: in ad hoc networks there are frequent changes in the topology, thus we can't use approaches like poison reverse.
We choose fresh routes over stale routes, we identify them thanks to sequence numbers that indicates the freshness of the communicated info. When a change occurs, the sequence number increases. The updates sent by nodes contain exactly these info (new sequence numbers, costs, freshness of routes).
Data broadcast includes multiple entries, each with:

- Destination address.
- Number of hops to reach a destination.
- Sequence number of the information received regarding the destination.

In the header, the data broadcast also include:

- Address of the sender of the message.
- Sequence number created by the transmitter.

There are two type of updates: full dump or incremental; this is done to decrease bandwidth consumption.
The costs represent the number of hops, our goal is to use fresh routes as short as possible, thus a link cost changes from 1 to infinite and from infinite to 1 (ma che cazzo vuol di'?).
We can detect a broken link or at layer 2:

- No hello messages received for some time.
- Retransmission attempts exceeds the MAC threshold.

Or at layer 3:

- Some nodes don't receive periodic updates by a certain neighbour.

Link cost increase $1 \rightarrow \infty$:

- The nodes incident to the link (A, B) discover it.

- Routes going through that link get assigned ∞ cost in the routing tables of A and B.
- A new sequence number is generated by the mobile node. Mobile nodes that are not the destination uses odd SN (numeri di sequenza dispari) and the destination uses even SN (pari).
- Updates with routes with infinite cost are immediately transmitted by nodes.

Link cost decrease ($\infty \rightarrow 1$):

- Immediately transmits updates.

When a node receives updates it sees if the costs can be improved. Routes with smaller sequence number are used, and if the SN is the same we use the shortest. Newly recorder routes are scheduled for immediate advertisement by link cost decrease. As soon as a route cost changes, the node may delay informing its neighbors, but immediately starts using the new information for its forwarding.

Correctness a distance vector algorithm is correct if no loop can occur. Assume that G(X) denotes the route graph from sources to a destination x and has no loops (assumption); a change occurs in i in two cases: the link from i to its parent P(i) broke and must be put to inf (a loop can never occur in this setting), or if i receives from a neighbor k a new route with a higher sequence number or equal sequence number but a smaller number of hops. If the new route has a higher sequence number and is thus chosen, it can't contain a loop, because if it does, the (new route) sequence number would be smaller than the one already stored in i [why? i don't know and the professor's explanation is a riddle]. Meanwhile if the new route has a smaller number of hops, it can't contain loop because distance vector algorithms always maintain loop-free paths (and this is true with the assumption that link weights are static or decreasing).

7.4.2 Optimized Link State Routing (OLSR)

It is a link state protocol for MANETs, it is suited for large and dense ad hoc networks. Its key concept is to decrease the overhead of flooding identifying a subset of nodes (multipoint relays) in charge of forwarding the info during the flooding process.

- *Multipoint relay* Y: a node selected by at least one of its 1-hop neighbours (say X) to relay all valid broadcast info received from X. $MPR(X)$ is the set of multipoint relays of the node X. The neighbours of X not in $MPR(X)$ receive and store the broadcast messages transmitted by X without re-transmit them.
- *Multipoint relay selector of* Y: a node X which has selected Y as multipoint relay.

It requires only partial link state to be flooded. Declare the links from MPR to their selectors is enough to ensure that routes to each destination can be found.
The protocol is fully distributed. Routes are always available when needed (proactive). Other features:

- Time between updates can be tuned to increase reactivity to topological changes.
- Doesn't require reliable transmission:
  - It can tolerate some losses due to needed info being periodically transmitted. Thus, OLSR control packets are embedded in UDP datagrams.
  - Sequenced delivery of the messages is not needed because nodes can reconstruct the full sequence properly thanks to sequence numbers.

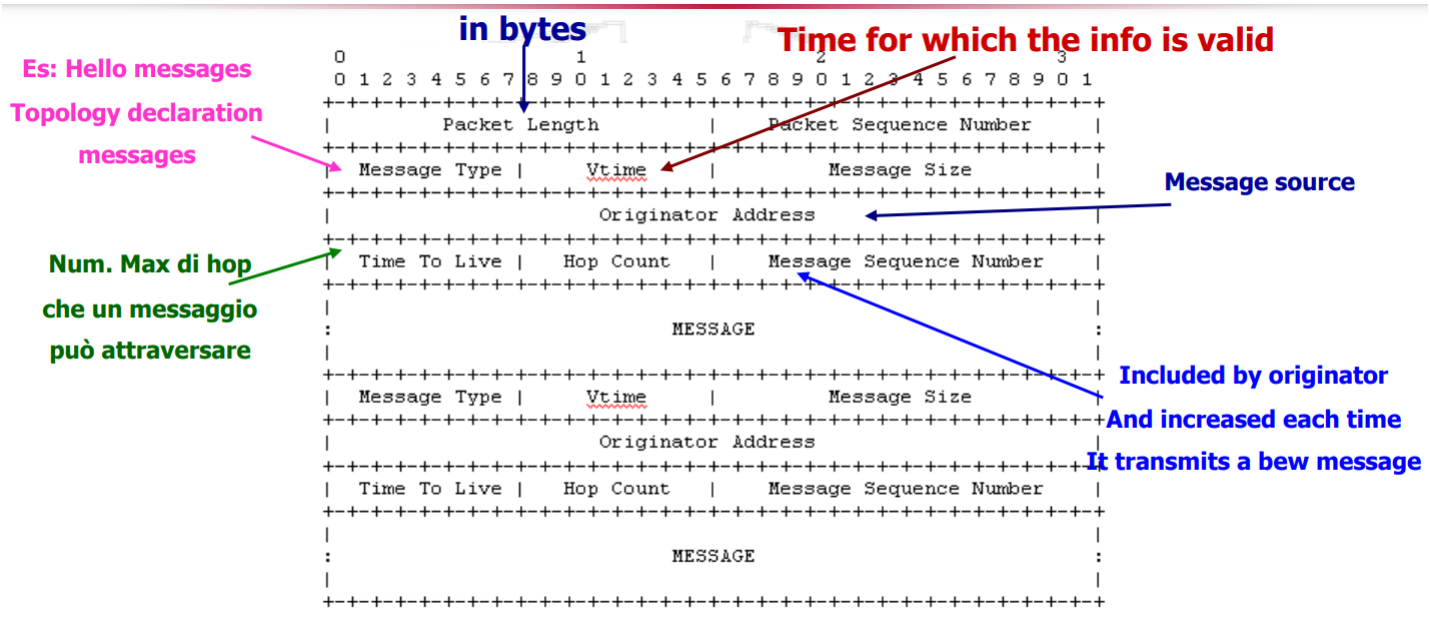


Figure 19: Packet format.

- Support to other MANET related issues, such as sleep mode operation and multicasting.

Hello messages are used to:

- Verify if links are up and running (sensing). If no hello message is received by a neighbour in a time interval, a timeout occurs and the link is considered down.
- To exchange info with neighbours . It allows to compute two hop neighbourhood, that is needed to select multipoint relays.

How do we select MPRs? Each node X selects its MPRs among its one hop neighbours. This set covers node X 2-hop neighbourhood. $MPR(X)$ is an arbitrary subset of node X 1-hop neigbours in such a way that each node Z in node X 2-hop neighbourhood have a neighbour in $MPR(X)$. The smaller is $MPR(X)$ the less control overhead is exchanged.
When receiving a message m at node Y:

- If the received message is not a duplicate, is valid and has a TTL that's not 0:
  - If it's received by an MPR selector of Y:
    - Retransmit m.
    - Reduce by one the message TTL.
    - Increase by one the message hop count.
    - Broadcast on all node on which Y interfaces.
    - Update or create the entry for the message in the duplicate set.

Information on topology are sent on every network node exploiting the backbone of multipoint relays and limiting as much as possible the amount of info. Each node can locally run a shortest path algorithm and then fill a routing table. The forwarding of a data packet is performed accordingly the routing table.

7.4.3 Proactive Approach Limits

Proactive protocols are costly in terms of overhead. The cost of maintaining routes updated doesn't have so much sense if we are in a context with medium-high mobility and dynamism (awake/asleep states). If in the network the traffic is generally low, does it really make sense to spend so much to maintain routes always updated?

7.5 Reactive Approach

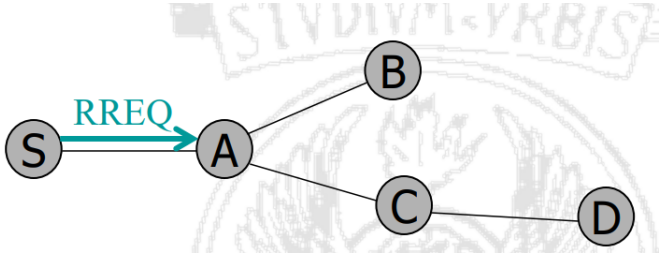
With "reactive" we mean that the source builds routes *on demand* by flooding. Characteristics:

- Maintain only active routes
- Route discovery cycle
- Typically, less control overhead, better scaling properties
- Drawback: route acquisition latency
- Example: AODV, DSR

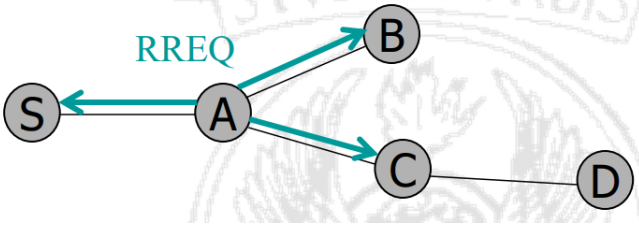
7.5.1 Ad hoc On-Demand Distance Vector Routing (AODV)

Nodes don't maintain routing info and don't do any routing table exchange. A node needs to discover and maintain a route to a destination only when it has to send a message to it. *Route discovery cycle* is used for route finding. Sequence numbers are used to prevent loop and as a criterion for route freshness. This routing protocol comes from DSDV but follows a reactive paradigm (pure on-demand operations). Provides both unicast and multicast communication.

Route discovery We can divide the process into points:
Node S needs a route to D AND does not have routing info for it in its table.



1. Node S needs a route to D.
2. Creates a Route Request (RREQ): enters D 's IP addr, seq#, S 's IP addr, seq# hopcount (=0), broadcast ID.
3. Node S broadcasts RREQ to neighbors.
4. Node A receives RREQ:
   (a) Makes reverse route entry for S dest=S, nexthop=S, hopcnt=1,expiration time for reverse path Source node, Source node SN,D,broadcastID also maintained.
   (b) It has no route to D, so it rebroadcasts RREQ (hopcount increased).
   (c) If it has already received that request (same source and broadcast ID) it discards the RREQ.



(d) if it knows a valid path to D it will send back a reply to the source

5. Node C receives RREQ:

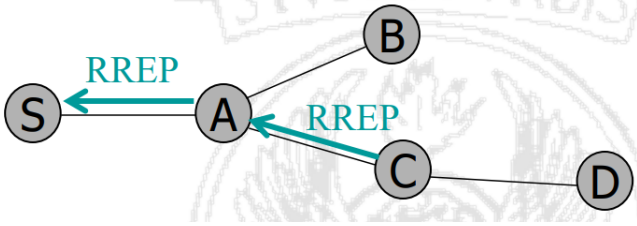
   (a) Makes reverse route entry for S dest=S, nexthop=A, hopcnt=2.

   (b) It has no route to D, so it rebroadcasts RREQ.

   (c) It has a route to D, and the seq# for route to D (is >= D's) seq# in RREQ.

   (d) C creates a Route Reply (RREP) Enters D's IP addr, seq S's IP addr, hopcount to D (= 1), lifetime of the forward route

   (e) Unicasts RREP to A



6. Node A receives RREP:

   (a) Makes forward route entry to D dest = D, nexthop = C, hopcount = 2.

   (b) Unicasts RREP to S.

7. Node S receives RREP:

   (a) Makes forward route entry to D dest = D, nexthop = A, hopcount = 3.

   (b) Sends data packets on route to D.

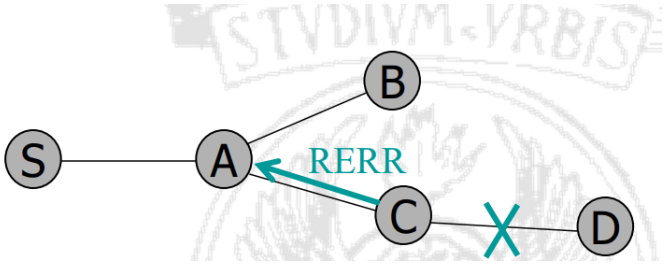
   Also the latest SN of the destination is updated when receiving the RREP. Nodes not along the path determined by the RREP will timeout after ACTIVE_ROUTE_TIMEOUT (3000ms) and will delete the reverse pointer.

If a node receives other RREPs for the same request it updates its routing info and propagates RREP only if this contains either a greater destination SN or same SN with smaller hopcount. Each node maintains the list of active neighbours, this is useful for route maintenance. The entries in the routing table are:

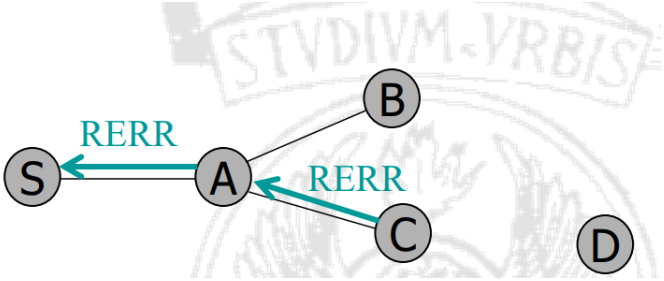
- Dest.
- Next hop.
- Hopcount.

- Dest SN.
- Active neighbors for this route.
- Expiration time for route table.

Route maintenance



1. Link between C and D breaks.
2. Node C invalidates route to D in route table.
3. Node C creates Route Error (RERR) message:
   (a) Lists all destination which are now unreachable.
   (b) Sends to upstream neighbours.
   (c) Increases by one the SN of the destination.



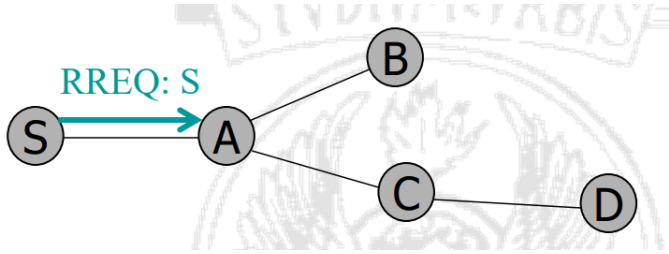
4. Node A receives RERR:
   (a) Checks whether C is its next hop on route to D.
   (b) Deletes route to D.
   (c) Forwards RERR to S.
5. Node S receives RERR:
   (a) Checks whether A is its next hop on route to D.
   (b) Deletes route to D.
   (c) Rediscovers route if still needed (in that case it sends a RREQ with a SN which is equal to the last known destination Sequence Number + 1).

Optimizations *Expanding Ring Search* prevents flooding of network during route discovery. The control Time To Live (TTL) of RREQ is used to search incrementally larger areas of network. We have the advantage of less overhead when successful, but longer delay if route not found immediately.
Local Repair allow to repair breaks in active routes locally instead by notifying. Uses small TTL because the destination has probably not moved far. If the first repair attempt is not successful, sends RERR to the source. W have the advantage of repairing links with less overhead, delay and packet loss; but we have longer delay and greater packet loss when it is unsuccessful.

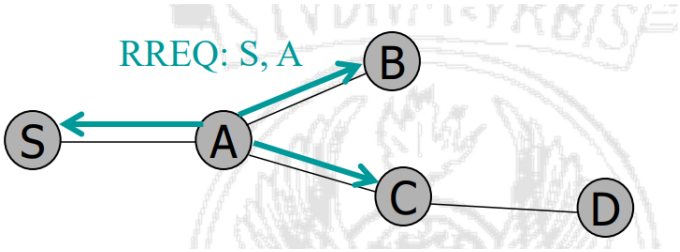
7.5.2 Dynamic Source Routing (DSR)

Like AODV it uses route discovery cycle for route finding and maintains active routes. It also uses *source routing*.

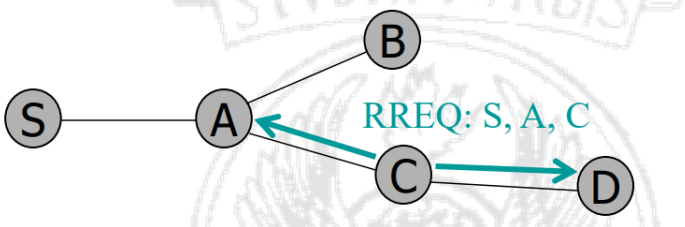
Route discovery



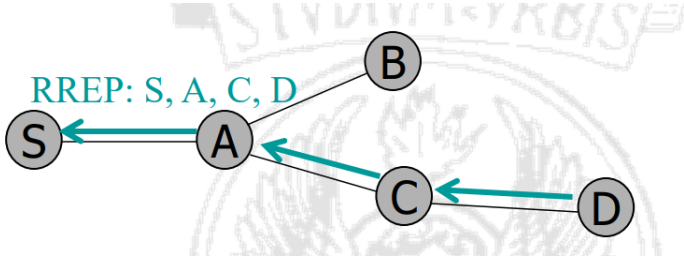
1. Node S needs a route to D.
2. Broadcasts RREQ packet.
   (a) RREQ identifies the target of the route discovery, contains a route record in which the traversed route is accumulated, contains a pair <initiator, request id> uniquely identifying the request.
3. Node A receives packet, has no route to D AND is NOT D, thus it rebroadcasts packet after adding its address to source route.



4. Node C receives RREQ, has no route to D, thus it rebroadcasts packet after adding its address to source route.



5. Node D receives RREQ, unicasts RREP to C. It puts source route accumulated in RREQ into RREP.



6. Node C receives RREP, unicasts to A.
7. Node A receives RREP, unicasts to S.
8. Node S receives RREP, uses route for data packet transmissions.

When a node n receives RREQ:

- If the pair <initiator address, request ID> has recently been seen, `discard`.
- If the node ID is already listed in the source route, `discard` to avoid loops.
- If n is the destination, send a RREP.
- Else append n ID in the source route and rebroadcast.

Route maintenance The two endpoints of a failed link are transmitted to the source in a route error packet. Upon receiving a RERR packet, a node invalidates all the routes that go through that link. If the route is invalidated but needed, a new route must be discovered.

Optimizations We can use caching widely to make node know all the routes to intermediate destination. When a node founds out a better route it improves its information. Transmitting packets and sending replies makes a node learn routes. A node that already knows the route to a destination (i.e., has the source route in its cache) can immediately answer a RREQ.
We can use *promiscuous mode* and make nodes discover new routes because neighbours nodes transmit them.
Minimization of RREQ overhead by setting TTL $= 1$ at first and then set it to ∞ if the node gets no answer.
The use of *path shortening*, i.e. when a node Y receives a packet from X but it has the source route $X, B, ..., C, Y$ reports that the path can be shortened.
Exponential back-off to decrease the quantity of RREQ sent if the network is disconnected.

7.5.3 AODV vs. DSR

- DSR uses source routing - AODV uses next hop entry.
- DSR uses route cache - AODV uses route table.
- DSR route cache entries don't have lifetimes - AODV route table entries have lifetimes.

7.5.4 Proactive vs Reactive - Drawbacks

Proactive drawbacks There is overhead due to updates, especially in presence of high mobility. We have large routing tables and low scalability. There is no real need to maintain a consistent view of the network topology all the time.

Reactive drawbacks The discovery phase introduces long delays. Both route discovery and maintenance are sensitive to node mobility. Route caching is memory greedy. The size of the header of a data packet can be hard to carry in DSR and this leads to low scalability. Operating in promiscuous mode and relying on flooding to discovery routes is energy consuming.

7.6 Geographically Enabled Routing

To try to overcome the problems of proactive and reactive approaches we find a new way of naming/locating the destination node. This new approach results in *geographic routing*. We have two main protocols: DREAM and LAR. There's the introduction of geo-enable routing costs because a node needs to know where it is and where is the destination.

7.7 Location-Enabled Ad Hoc Routing

Nodes are equipped with positioning system devices (GPS receivers) to make them aware of their position. This gives us the possibility to do directional routing.
The strengths are:

- No need to update big routing tables or piggyback routes to the packet because the destination positions has to be known at the source.
- No need to know the nodes on the way to the destination.

The drawbacks are:

- There is the need of extra hardware.
- It is hardware-dependant (it may have some limitation).
- It does not solves scalability issue.

7.7.1 Location Aided Routing (LAR)

It exploits location info to limit the range of RREQ flood. It has the concept of *expected zone*, i.e. the area that is expected to contain the location of the destination. This is determined based on the old location and knowing the destination speed. RREQs are limited to a *request zone* that contains the expected zone and location of the sender ndoe.

LAR-1 The request zone is the smallest rectangle that includes both the source location and the expected zone. Only nodes within the request zone forward RREQs, that explicitly specify the request zone. Each node must know its physical location to determine if it's in or out the request zone.
If a route discovery attempt goes wrong, the sender can enlarge (after a timeout) the request zone to start another route discovery. The rest of the route discovery protocol is similar to DSR.
Talking about LAR routing, initially we assume that a node Y only finds out location info for node X during route discovery. This info is used for a future route discovery:

- Updates on the node position are piggybacked in the RREP message.
- This allows to reduce overhead associated to route discovery.

The destination may also proactively distribute its location info, but in this case the control traffic for geographic info updates could be high. This issue will be later solved with DREAM.

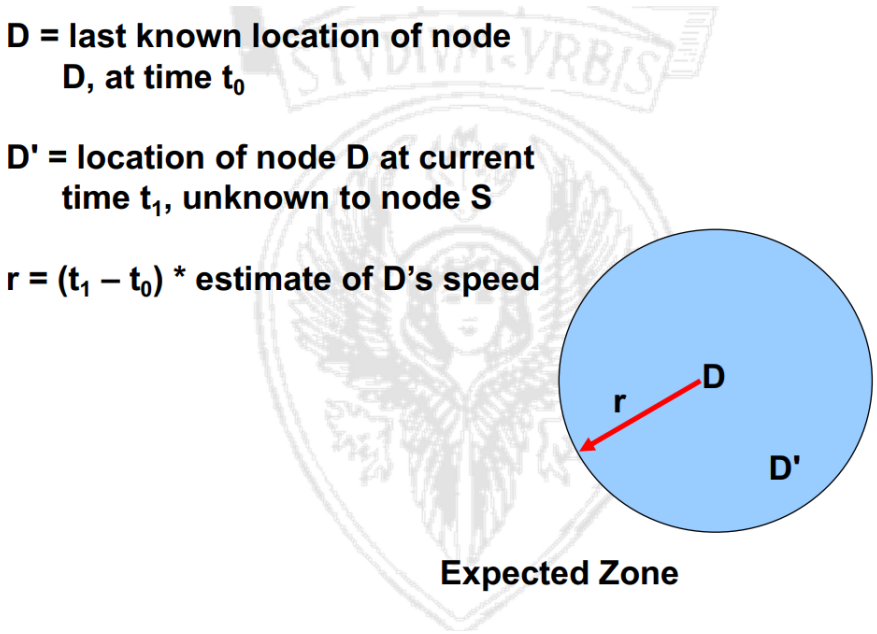


Figure 20: LAR Expected Zone.

LAR-2 Each node relays RREQ if it is closer to the destination than the source (greedy forwarding).

7.7.2 Distance Routing Effect Algorithm for Mobility (DREAM)

Source S determines the location of destination D at time t_0 based on its location table. Based on the current time t_1 and t_0, S determines the area in which D can be found (D's direction). S transmits the data packet to all of its neighbours in D's direction and then each neighbour does the same until D is reached.
Obviously this approach needs to update the location of a node periodically. This is done via a broadcast of location info and we need to know how far each location packet should travel and how often should it be sent. Faster the node, more updates it sends.

Location effect Closer nodes look like they are moving faster, thus we need to receive more location updates from closer nodes. Each location packet is associated with an age that says how far the packet must travel.

Strengths First of its kind and robust (multiple route to the destination).

Drawbacks Even if directional, it's still a flooding protocol. It's not so scalable due to the periodic updates.

7.7.3 GeRaF

It integrates geographic routing, awake/asleep schedule and MAC protocol. The nodes alternate awake and asleep states following a duty cycle. Each node needs to know:

- Its location.
- The destination (sink) location.
- The location of the transmitter.

A greedy approach is used, trying to select relays in order to go as near as possible toward the destination at each hop.
How to make a contention-based routing work in the presence of sleep modes? MAC and routing used simultaneously, every node does not know its neighbours and their schedules and this leads to a low overhead. There's no need of routing tables because everything is based on the knowledge of the nodes location and the sink location.
An example of a GeRaF operation:

- RTS message invites all awake neighbours to become a relay.
- The nodes are split into areas depending on their position and the ones that are the nearest to the destination should win.
- Winning nodes answer with a CTS message:
  - If no node answer the transmitter polls (from *polling*) the next area.
  - If there is one answer it can send DATA.
  - If there are multiple answer there is a collision that will be solved thanks to MAC. The transmitter sends a collision packet.

To handle collision:

- A node receiving a collision packet will transmit a CTS with probability p if and only if it sent a CTS previously.
- Loop until only one node will send a CTS.

GeRaF is an example of a cross-layering protocol. Cross-layer design allows protocols belonging to different layers to cooperate and share network status information, this design choice is extensively used for IoT devices.
If all areas are polled unsuccessfully we have to try again after some time decided using an exponential back-off. Besides, GeRaF does not ensure that we will always reach the destination. This problem happens often in low density networks. We can set a max number of attempts

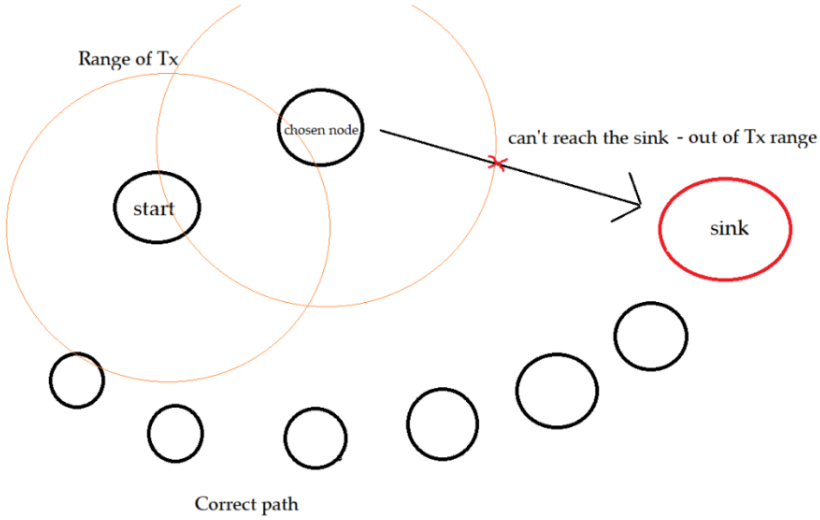


Figure 21: GeRaF not reaching the sink node.

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 and almost blacklisted. However, we may still have problems.

7.7.4 Adaptive Load-Balancing Algorithm (ALBA)

It handles dead ends using *Rainbow*, a node coloring algorithm to route around them. Results for high and low density of nodes.
The *geographic routing* paradigm follows the rule: "forward the packet to a node that offers advancement toward the destination". The pros of this approach is that is virtually *stateless* because it only needs nodes locations. The cons is that it requires positioning estimation and mechanisms to route packets out of dead ends.

Dead end problem If the routing algorithm is tuned to achieve a positive advancement at each step, dead ends may occur. Like we've seen before we may not arrive to a destination at all, and if we are not able to re-route that packet it'll be lost.

ALBA Approach Nodes forward packets in bursts (we send up to M_B packets back-to-back (consecutively)) with an adaptive burst length. The forwarders are elected based on:

- The ability to receive and correctly forward packets. The metric used are:
  - Queue Priority Index (QPI).

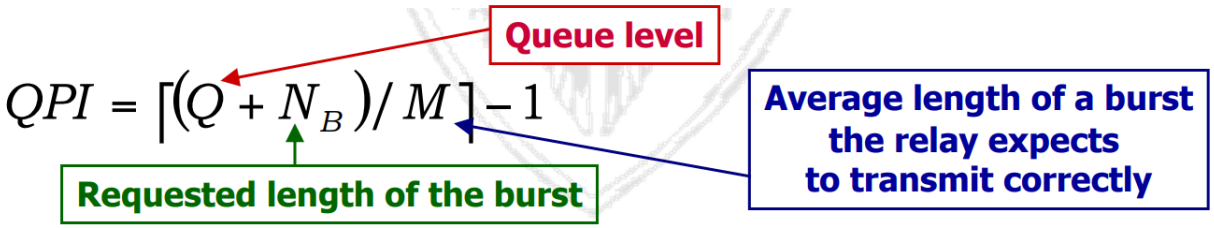


Figure 22: ALBA QPI.

  - History of the transmissions of the relay.
  - Number of packets the sender needs to transmit.
- The geographic proximity to the destination using Geographic Priority Index (GPI).

The metrics used for the relay choice ensures load balancing because it preferably chooses:

- Nodes with low queue, especially with high N_B.
- Nodes with good forwarding history (through M).

Relay selection phases:

1. Selection of the best QPI:
   - In the first attempt we search for $QPI = 0$, then for $QPI = 0, 1$ and so on.
   - Awaking nodes can participate in this phase.
2. Selection of the best GPI:
   - Performed if more than one node with the same QPI is found.
   - Awaking nodes can't participate to speed up the algorithm.

Dead ends may still happen!
An example: $QPI(A)$ is low ($M = 2$) but nearer to the sink then B, $GPI(A) = 1$, $GPI(B) = 2$. However, B has greater reliability (M) and comparable queue occupancy (Q) resulting in $QPI(B) > QPI(A)$. If B is sleeping at transmission time, node A is selected for its better GPI. (ma che cazzo di esempio è? ahahah)
ALBA deals with node contention in a similar way to GeRaF. If no node responds then the sender continues pinging the region, if more than one node responds a collision solving algorithm starts and when only one node sends a CTS the data transmission starts.

Collision Resolution Algorithm

- Upon receiving a $COLL_{GEO}$ message, nodes reply based on their GPI.
- Upon receiving a $COLL_{ST}$ message, nodes persist in sending CTSs with probability 0.5.
- Should they all decide to stay silent, the following $COLL_{ST}$ message enables a further decision.
- Eventually, the process ends with a single valid relay being selected.

Rainbow Algorithm We need a method to route around dead ends in low density network. Nodes that recognize themselves as dead ends progressively stop proposing as relays. To route traffic out of the dead end, they start to transmit packets backward (not advancing to the destination). This is done hoping that a relay with a forwarding path to the sink can eventually be found. To apply this algorithm we use a recursive coloring procedure.
Yellow nodes represents the one that exhibit a greedy path to the sink; initially, all the nodes are yellow. If a node recognizes himself as a dead end, then it will color himself with red, and will send back (negative advancement to the sink) its packet to red or yellow nodes. If red nodes can't advance nodes, they will color themselves as blue, and will search for blue and red relays to send packed forward (positive advancement to the sink). If even this way they can't find a route, then they will color themselves as violet and relay packets backward (in a negative advancement to the sink) to blue and violet nodes. There can be many colors, e.g. h, and each color has a label C_i; yellow nodes have always the C_1 label, and in general: odd-labeled nodes relay forward, even labeled nodes relay backward, and both relay to neighbors with the same label C_k or the previous one C_{k-1}, with the exception of yellow nodes, who relays only to other yellow nodes labeled C_1.
To sum up the algorithm:

- **Concepts**:
  - Nodes progressively realize to be dead end and automatically adapt to this condition.
  - More colors mean more nodes can successfully deliver packets.
- **Pros**:
  - Routes around dead ends.
  - Completely blind and distributed.
  - The load-balancing features of ALBA are used throughout.
- **Cons**:
  - The network requires some training for nodes to achieve the correct color.

7.8 Localization

It helps with some protocols and it's needed for being able to identify where an event occurs. We can't use GPS at every node for:

- High power consumption.
- Works only when in LOS with the satellites.
- Over kill, we often need only a relative position.

A basic step is to evaluate distance between two nodes and we have different techniques depending on the underlying hardware: AoA, RSS, ToA. We may also use range-free approaches that relies on the hops number between nodes to estimate distance.

7.8.1 Angle of Arrival (AoA)

Measure direction of landmarks. Simple geometric relationships can be used to determine the location finding the intersection of the lines of position. An example is radiolocation based on Aoa. This can be done using directive antennas and a compass and needs at least 2 measurements.

7.8.2 RSS and Time of Arrival (ToA)

Measure distance to landmarks or does ranging. An example is radiolocation using:

- Distance via received signal strength. It uses a math model that describes the path loss attenuation, each measurement gives a circle on which the MS must lie.
- Distance via ToA. Distance is measured thanks to propagation time, each measurement gives a circle on which the MS must lie. Both active and passive approach:
  - Active: receiver sends a signal that is bounced back so that the receiver knows the round trip time.
  - Passive: receiver and transmitter are separated and time of signal transmission needs to be known.

7.8.3 Multilateration

For a 3D localization, an additional anchor will be needed: two sphere intersecting will form a circle; a third one intersecting with the circle will give two points of uncertainty (like two circles intersecting together), so a fourth one is required. MSE between the estimated position and the anchor is the metric used to minimize the error when the position is evaluated. The practice where beacon nodes, helps the others to estimate their position is called multilateration; it is a simple solution that works for small networks, but suffers if errors begin to accumulate. The node that can understand their position become beacons and help others to know their position.

7.8.4 Range-free Localization

It doesn't use ranging but only info available. Anchors point know their position according to a common coordinate system, all nodes compute Dijkstra between them and anchors, the latter compute their relative distance through hops. In this way anchor A can say "I know where I am and where are the other anchors, thus I can estimate the length of one hop". This info is used to estimate every distance $D(x, y)$ where x is a node and y and anchor. Based on these distances every node computes its own position using triangolarization. It is not very accurate.

8 Energy-efficient MAC protocols

8.1 Performance goals

- **Energy efficiency**. We need to control as much as possible the sources of energy waste:
  - *Collisions:* they increase energy consumption causing re-transmissions and also increase latency.
  - *Overhearing:* the situation when a node is listening for a packet that it's not addressed to itself.
  - *Control packet overhead.*
  - *Idle listening:* the amount of time in which a node is waiting for a packet that won't be sent is the main source of energy consumption.
- **End to end latency.**
- **Fairness.**
- **Network capacity/scalability.**

8.2 Sensor Mac - S-MAC

8.2.1 Nodes sleeping scheme

Nodes follow an awake/asleep schedule with a given duty cycle. In certain protocols, such as S-MAC, nodes schedule are synchronized, thus they all transmit/receive in the same slot. Before a node starts its cycle, it needs to choose a schedule and inform its neighbours by broadcast. At start-up node x listens for some random time:

- If x receives a `SYN` message from a node y, it synchronizes to its schedule. x is called *follower*. It waits for a random delay and rebroadcasts its schedule.
- Otherwise, node x selects a random time T to sleep, then wakes up again and sends T to neighbours in a `SYN` message. x it's called *synchronizer*.
- If a node, that has already selected a schedule, receives a new one, it adopts both and broadcasts the newest.

Each node has also to maintain a *schedule table* that stores the schedules of all its known neighbours.
When a node has a packet to send it waits for the destination to be ON and sends the packet following CSMA/CA:

- It performs carrier sense for a random interval.
- If no transmission within this interval the floor is taken (physical carrier sense) to transmit RTS/CTS.
- If the RTS/CTS is successful (virtual carrier sensing) data is sent which is followed by an ACK.
- NAVs are used to decide for how long nodes should go to sleep before they can try to access again.
- To better exploit the time needed to handshake (RTS/CTS) bursts of packets are transmitted if more packets are in queue for the same destination.

Some initially exchanged SYN messages can be lost and this can cause clock drifts. Thus every node has to periodically send a SYN message.

8.2.2 Limits

It needs synchronization that adds control overhead. Throughput is reduced since only the active part of the frame is used for communication. Latency increases because when a node generates a packet it has to wait for the next hop relay before sending it.

8.3 Timeout MAC - T-MAC

An observation on S-MAC. We have 2 critical parameters, active time and frame time. A long frame time increases latency. Given an active time, the longer the frame time the lower the energy consumption. The active time should be chosen based on traffic, higher the traffic longer the active time. In S-MAC, however, these 2 parameters are fixed.
T-MAC fixes frame time but active time is dynamic.
The nodes synchronize their schedules using the S-MAC virtual clustering. The protocol uses CSMA/CA within an active time and packet transmissions in bursts. If there is no transmission from neighbours for a TA time, the active time is aborted and node goes to sleep. This is a change w.r.t. S-MAC. TA is reset if:

- Any data is received on the radio.
- Communication is sensed on the radio.
- Data are transmitted.
- RTS/CTS are exchanged by neighbours.

TA is decided taking into account the time for RTS + CTS + waiting time.

8.3.1 Differences w.r.t S-MAC

When a node sends an RTS but doesn't receive a CTS back this may be due to one of the following events:

- The RTS was not received due to collisions.
- The receiving node cannot answer due to an RTS/CTS overheard.
- The receiving node is sleeping.

In the first 2 cases it is wrong to reduce the active time, a node should retry to resend the RTS at least twice before going to sleep.
Early sleep may degrade throughput.

8.3.2 Early Sleep Problem

The node D goes to sleep before C can send an RTS to it.

Solution 1 The FRTS (Future RTS) packet exchange keeps the node D awake.

Solution 2 Full buffer priority: upon receiving an RTS, a node which has the buffer almost full, instead of answering with a CTS sends immediately an RTS.

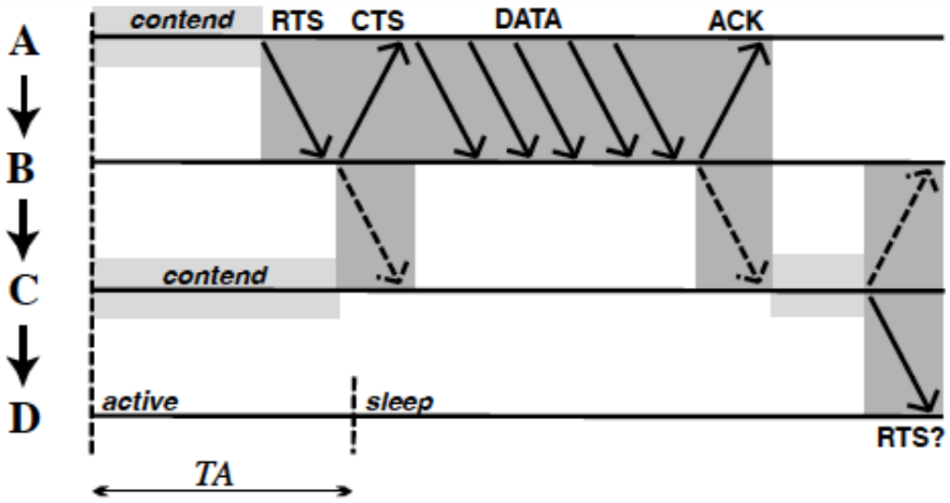


Figure 23: Early sleeping.

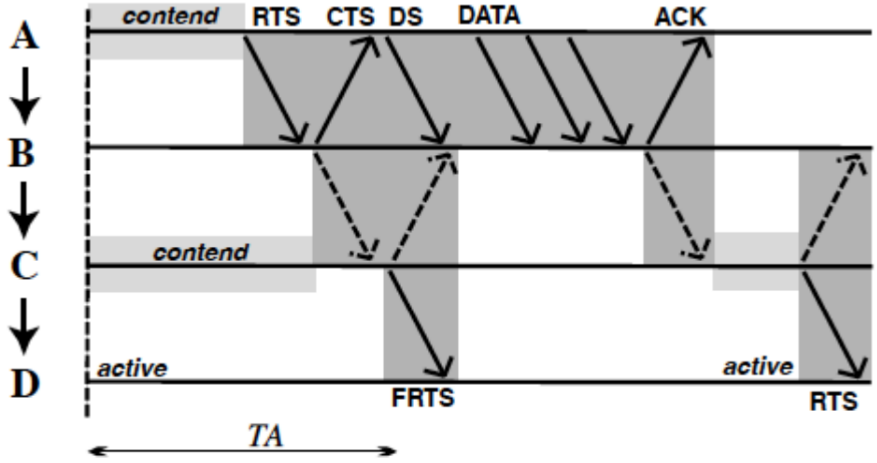


Figure 24: Early sleeping, first solution.

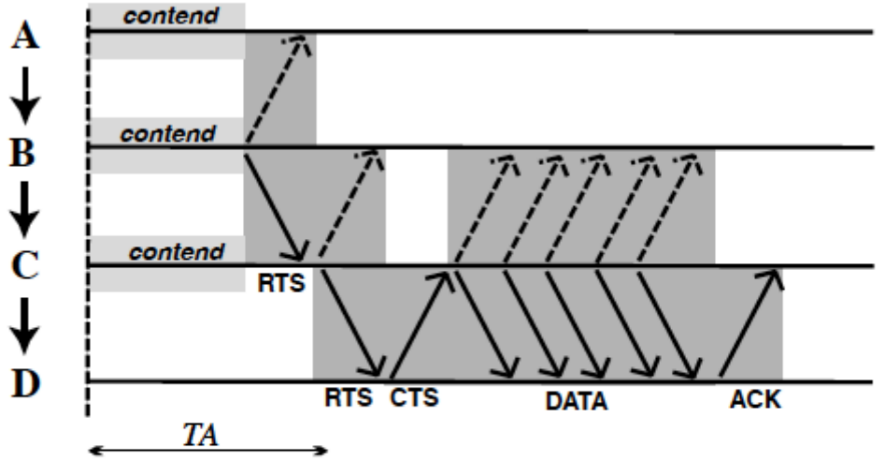


Figure 25: Early sleeping, second solution.

8.4 B - MAC - Berkeley Medium Access Control

For an effective collision avoidance, a MAC protocol must be able to accurately determine if the channel is clear (Clear Channel Assessment). BMAC proposes a way to estimate the channel noise to determine if it is free. The solution has been validated through experimental data.

Receiver side B-MAC duty cycles the radio through periodic channel sampling, called Low Power Listening (LPL). Each time the nodes wakes up it turn on the radio to check for activity. If the node detects some, it chooses to stay awake for the time required to receive a packet. After reception / timeout the node goes back to sleep.

Transmitter side The sender transmits a preamble, then the data. To reliably receive data, the preamble length matches the time that a node check for activity on the channel (preamble long enough to be heard in the time a node does LPL). A key advantage of this protocol is to

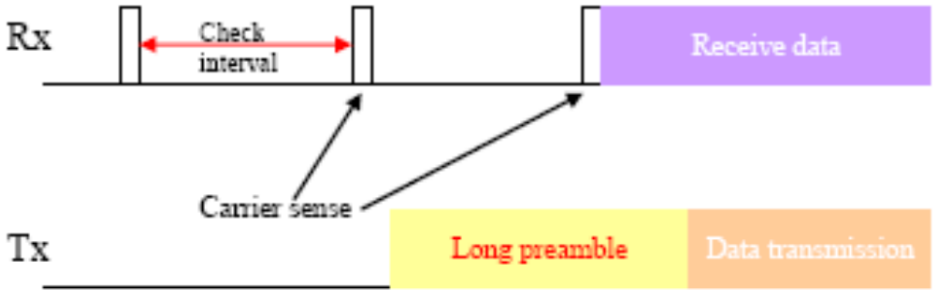


Figure 26: B-MAC.

be completely asynchronous, because nodes doesn't have to bind their duty cycles. However, long preambles lead to performance degradation. Remember that the receiver wakes up at the start of the preamble and stays awake until the data/ack exchange is terminated. Besides, also nodes that are not involved in the communication but sense the preamble have to be awake until timeout.

8.5 X-MAC

The idea is to enhance the preamble in two ways:

- Embed address info in it to avoid overhearing (i.e. being sensed by uninvolved nodes).
- Use a strobed preamble, i.e. a series of short preambles with pauses between them. When the sender receives an ACK stops sending them.

8.6 WiseMAC

It is based on the idea to mix the strengths of synchronous and asynchronous approaches. We don't want to keep the nodes synchronized and to know when a node will wake up, thus we use X-MAC as our base approach. However, when we start transmitting we have very likely to transmit more than just one packet. Thus it's useless to send preambles every time we send data, instead we inform our neighbours on our duty cycle to tell them when will be the next time 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, even if we are asynchronous we are somehow informed on the duty cycle of the nodes.

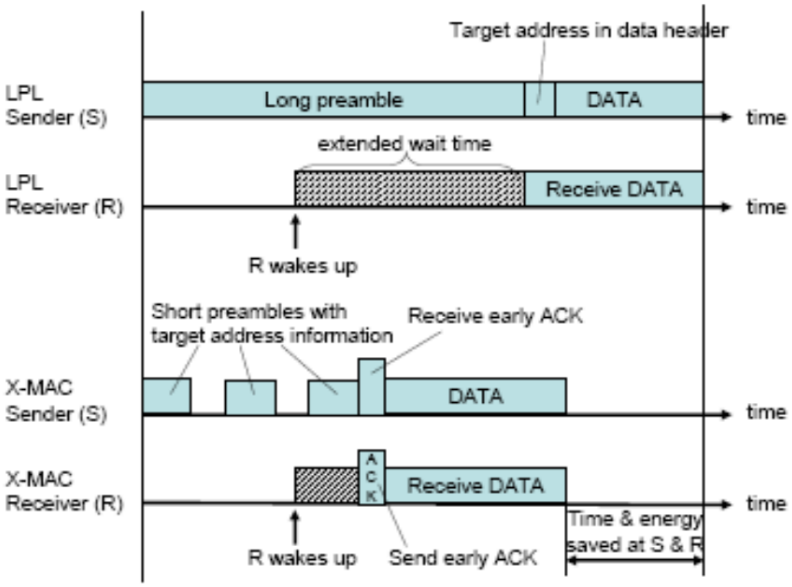


Figure 27: X-MAC vs. B-MAC.

9 IoT Standardization

9.1 6LoWPAN

9.1.1 Protocol stack

The protocol stack presents a few differences with the Internet one. The layers are the same but we use different standards.

- *First layer - Physical:* we have IEEE 802.15.4. We have dedicated frequencies on which transmit.
- *Second layer - Data Link:* we have MAC protocols defined by IEEE 802.15.4.
- *2.5 layer - Adaptation:* it is the LoWPAN itself, here we optimize and compress headers to make addressable the smart IoT devices with IP, but decreasing any extra overhead. Thus, we can't use the traditional TCP/IP protocol because it will be too energy consuming.
- *Next layers:* having made an adaptation lets us run the traditional IPv6. We will use an optimized UDP/IP protocol that allows end-to-end communication with traditional non-IoT IP devices.

9.1.2 Physical Layer Packet Structure

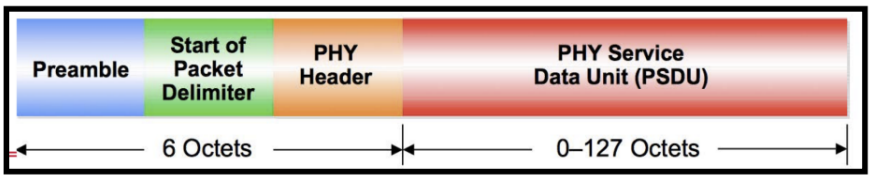


Figure 28: 6LoWPAN Physical Packet.

- The preamble is needed for synchronization.

- The Start of Packet Delimiter tells us the start of the data unit.
- Physical layer header.
- The PSDU is the actual data field.

It'd be good if the bands allocated all over the world were the same, but it's true only for 2.4 GHz (end 5 GHz I assume) and not for 868MHz (only Europe).

9.1.3 Network topology

We know that we have to handle every kind of device. There will be some that aren't even able to act as a relay and others that are far more powerful. This and other reasons raises up the need to have a **PAN Coordinator**. Its tasks are:

- Net ID assignment.
- Frequency selection.
- Handling request to join.
- Packet relaying.

We have three main types of topology: stars, tree and meshes.

- **Star:** everyone is connected directly (wireless) to every node through the PAN Coordinator, that acts as the only relay.
- **Mesh:** There is a PAN Coordinator but it's helped by other nodes that can act as a relay, building a multi-hop infrastructure.
- **Tree:** we have one PAN Coordinator and other "sub" Coordinators in a hierarchical network. This topology acts like the Mesh, but every relay node is a Coordinator and is responsible only for a group of nodes.

The PAN Coordinator will have to pick the channel to use among the available ones and also to handle the network topology upon nodes that come (request to join) and go away (request to leave). In a distributed system we have to elect a PAN Coordinator and I have a certain feeling that Paxos will be used, lol. Mei, I will always love you.

9.1.4 Data Link Layer (MAC) Packet Structure

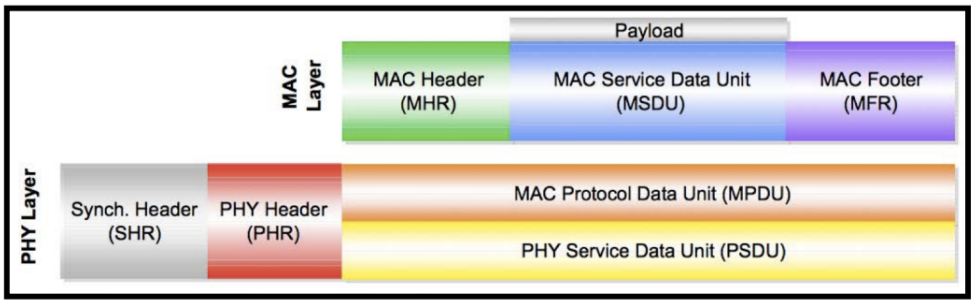


Figure 29: 6LoWPAN MAC Packet.

This MAC packet data unit is basically the payload of the physical layer; so it passes to the physical layer, that has also the synchronization header and the physical header.

9.1.5 Collision Handling

We can operate in two different ways.

Beacon Mode In this case there is a central unit (i.e. PAN Coordinator). The PAN Coordinator instructs nodes about the start of some time interval. Periodically, the PAN Coordinator will send a beacon saying who has something to receive. Then, according to a CSMA/CA approach, nodes compete for the uplink communication. Through a Beacon Extension Period we can also inform devices that they have special dedicated time slots called Contention-Free Resources.

CSMA/CA Beacon-less Mode If we don't have a central access point (e.g., ad-hoc networks) we can operate in Beacon-less Mode. Here we support different kind of frames of the data unit:

- **Data frames:** for the transport of actual data.
- **Acknowledgment frames:** meant to be sent back by a receiver after a successful reception of a data frame.
- **MAC layer command frames** used to enable various MAC layer services.
- **Beacon frames**: used by a coordinator to structure the communication with his associated nodes.

9.1.6 Adaptation Layer

To make smart objects IP addressable we need an adaptation that is part of the 6LoWPAN activity. Smart objects are permanently identified by EUI-64 identifiers (8 bytes), while IPv6 addresses are longer (128 bytes). In general, both of the addresses are too long to be communicated because every time we have to communicate to/from device, we have to include them in the headers. So we would like to allocate a local address as part of our network, which is a 16 bits address. All the rest would be a shared sub-net mask which is part of the network; but that is something that will be used only by the edge router to make a full IPv6 address when communicating with the external network.

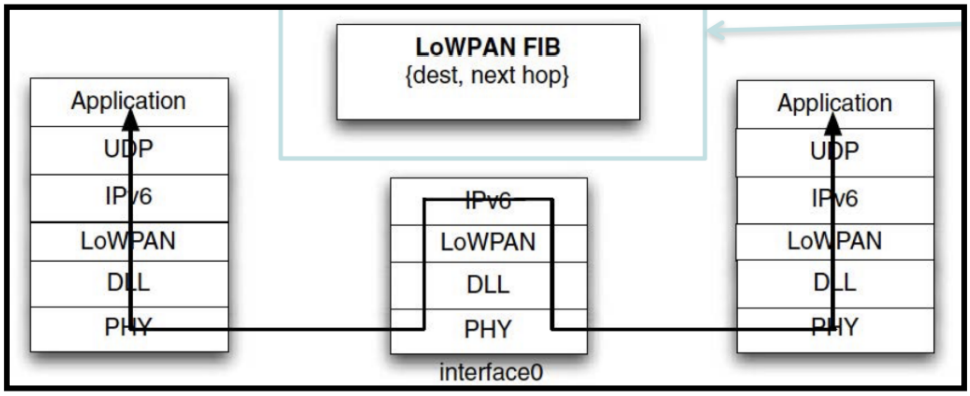


Figure 30: 6LoWPAN end-to-end communication.

9.1.7 Fragmentation

It is used when transmitting Packet Data Units at layer two and three larger than 128 bytes. The fragmentation and the reassembly is performed at the link level. We make use of a **fragmentation header** that, together with source and destination, is used to identify the original packet by identifying the order of fragments. The fragment header composition:

- **Datagram Size:** describes the total non-fragmented payload.
- **Datagram Tag:** identifies the set of fragments and is used to match fragments of the same payload.
- **Datagram Offset:** identifies the fragment's offset within the non-fragmented payload.

9.1.8 Mesh Addressing

In case of Mesh Addressing Header, we need to provide info on source and destination addresses and the maximum number of hops that we can traverse. We can avoid using the IP addresses and we'll use the 16 bits ones. Two kinds of routing are supported:

- **Mesh-under:** this uses the layer 2 addresses to forward data packets.
- **Route-over:** this uses the layer 3 addresses (IP).

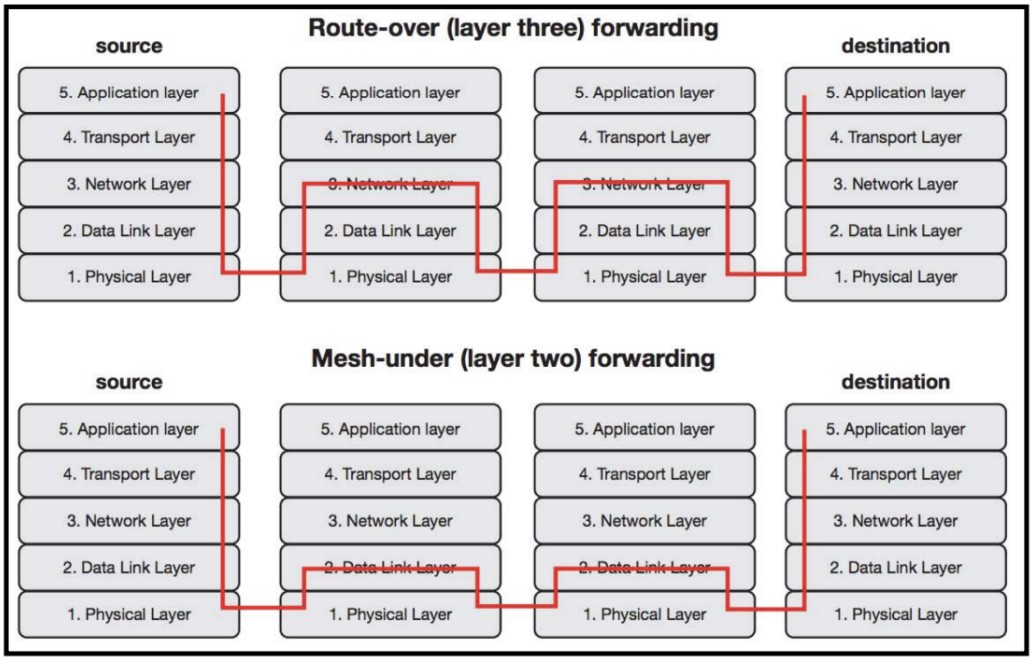


Figure 31: Mesh-under vs. Route-over.

9.1.9 IPv6 Header Compression - HC1 and HC2

Typically the IPv6 header has a number of field such as: protocol version, traffic class, flow label, payload length, next header, hop limit, source and destination addresses.
We don't have to transmit the version because we know it'll be IPv6. The value of traffic class and flow label is typically 0. Thus we can use a special boolean flag C (0,1) to understand if we can skip these two values ($C = 1$ to skip).
The payload length can be retrieved by other headers, thus we don't need to transmit it. We

same other assumption also with the next header field, assuming it'll be one among TCP, UDP and ICMP and hence encoding the field with only 2 bits. Besides, the source and destination addresses can be reduced to the 16 bits ones saving other space.
This method is called **HC1 Compression**. In the HC1 header there's a field that, if set to 1, tells us the next header will be UDP. This is useful because we can compress UDP headers. We know that we have to specify source and destination port to perform multiplexing. What we do is reduce the subset of the possible ports to reduce the size of the field.

9.2 IoT Network Bootstrap

The edge router broadcasts general information related to addressing a node-router association (choose the router responsible for a certain group of nodes). The edge router will assign local addresses caring about not generating duplicates.
If a new device joins the network it performs an active scan asking to the edge router to present itself with a Router Solicitation. The edge router answers to the request with a Router Advertisement, now the new node can join.

9.3 Collection Tree Protocol - CTP

It has been designed to support loss in low power IoT devices. The protocol anycasts routes to the sinks, in this way we collect data from the network to a small part of it (roots or BSs). Each node selects one of its neighbours node as a **parent**, every parent acts as a relay "pushing" the node towards the sink.
CTP is a distance vector protocol and the metrics to select the next hop are:

- Distance in hops from the sink.
- Quality of the local communication link.

The protocol wants to pursue the following properties:

- **Reliability:** at least 90% of the packets are delivered correctly.
- **Robustness:** it should be able to operate without tuning or configuration in a wide range of network conditions.
- **Energy efficiency**.
- **Hardware independence**.

9.3.1 Parent Selection

We use a Distance Vector approach to select the best parent based on a specific metric. This metric combines 2 aspects:

- We would like to go through the shortest path, thus we take hops into account.
- We would like to re-transmit packets as little as possible, thus we take link quality into account.

The metrics that combines these two together is called **ETX** and is the expected number of transmissions from a node to the sink. We compute ETX locally at each node by looking at 1-hop neighbour and choosing the one that provides the best ETX.
How can we do this? We have 2 kind of information transmitted in CTP networks: control and data. The control information are beacons (Prof. Petrioli means "beacon messages" and not

physical beacons) generated periodically by the sink and shared in the network. These beacons help us to build a tree structure allowing us to deliver to the nodes the ETX information. When we have to re-construct the structure we just use another beacon exchange.
When we perform this beacon process we have a given number of beacons to transmit per unit of time and a sequence number in data packets. With these values we can estimate the percentage of packets that we will receive and this lets us compute the link quality. When we receive the information about the ETX from our neighbours we tweak the value also considering link quality. The parent selection is performed only among non-congested nodes.
When we transmit data we have 2 flags, P (pull flag) and C (congested flag), the sequence number and a THL (Time Has Lived). We put the estimated ETX in the packet to allow us to identify it if it goes back to us (loop). If there is a loop, we set P to 1 and force the re-computation of routes. The units that implements what described in CTP are called **Control Plane** and **Data Plane**.
The Control Plane implements the distance vector logic using a *Link Estimator* that provides info to compute ETX locally. There is an exchange of ETX between neighbours and every node can compute locally, with Bellman-Ford, the best parent based on ETX metric. This allows the ndoe to set up a *Forwarding Table*.
When a node has selected the parent it will forward packets into the *Forwarding Engine*.

Loop Detection We know we embed ETX also in Data Packets to use the value to identify a loop, but how can we do it?
Let's consider the situation in the following images: A will select B as parent and B will select

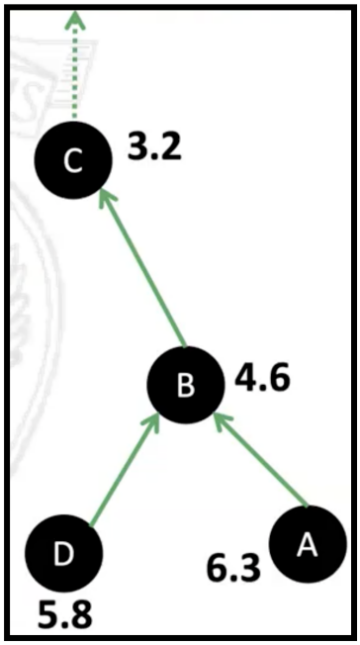


Figure 32: Loop Detection in CTP, 1.

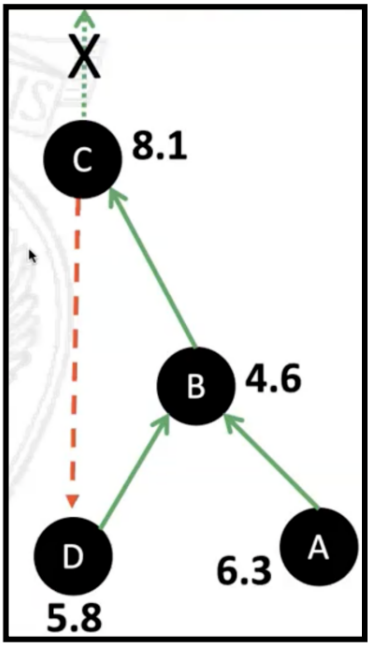


Figure 33: Loop Detection in CTP, 1.

C. If the link from C to the sink breaks, C has to choose another route and chooses D, raising the Count to Infinity problem by creating a loop. This is not only a loop in evaluation, but also in transmission, because every node forwards its packets only to its selected parent.
If we look at the property of ETX we can easily understand that, as a packet travels from node to node, it has to monotonically decrease. Let's reconstruct the process; node A sends a packet to B with $ETX = 6.3$, B with $ETX = 4.6$ to C, C to D with $ETX = 8.1$.We know we will go into a loop. but D doesn't know it yet. Infact, it's normal to receive an ETX higher than the one computed locally, thus D forwards the packet to B with $ETX = 5.8$ and B does the same to C with $ETX = 4.6$. At this point C can look at ETX and see that it's smaller than

the one computed locally. In this way C understands that there's a loop and sets the Pull flag to 1 asking to recompute the routes to the sink.

9.3.2 Beaconing Process

It is the process of sharing a beacon message contain info on the network from the sink to every node of the network. This approach has a cost because the nodes aren't always sending data packets but has to stop to send beacon messages. CTP defines when the beaconing process has to take place. The idea is that we do a beaconing process at the beginning. After some time we re-do the process but, if everything is going fine (no broken link, no loops, ...) we double the interval between two beaconing process (i.e., the first at time 0, the second at time 2, the third at time 4, the fourth at time 8, etc...). When in the network starts to travel a packet with a Pull flag set on, this means that something went wrong and a beaconing process is triggered, resetting the increment. This idea is called **Trickle Algorithm**. On average, CTP can successfully delivery 98% of the packets!

9.4 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 guaranteed real time exchange, resilience to interference and ability to increase capacity. In order to fulfill these needs IEE 802.15.4 has released an extension in 2016 and it's working even now on improve the standard.
It is based on a CSMA/CA approach. Limits:

- No warranties on delay.
- Does not support frequency hopping, thus can still have interference.
- Not the ideal MAC in high-traffic scenarios.

The 2016 extension brought some improvements such as:

- **Low Energy (LE):** gives to devices the possibility 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:** changes association procedure which was trading off energy for latency in case of applications which require fast association.

There are some **variants**:

- **Radio Frequency Identification - BLINK**: it supports very simple applications where we simply have to provide an ID. The exchange of IDs is for the sake of item/people identification, location and tracking.
- **Asynchronous multi-channel adaptation - AMCA**: supports dynamic multi-channel use in distributed beaconless networks.
- **Deterministic and Synchronous Multi-channel Extension - DSME**: supports time-critical application for large networks with beacon PANs. It enhances the basic Guaranteed Time Slot (limited to 7 slots per frame). DSME uses a multi super frame and multi-channel operations.

- **Low Latency Deterministic Network - LLDN**: designed to support commercial application requiring low and deterministic latency. There are a large number of sensors/actuators monitoring and controlling an operation and this guarantees a latency below 10 msec. There is also a multi-channel extension enabling the PAN coordinator to simultaneously receive/transmit over multiple transceiver (instead of only one).
- **Time Slotted Channel Hopping - 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.

9.4.1 Time Slotted Channel Hopping - TSCH

It is a channel access method for shared-medium networks. Every node synchronize on a periodic slotframe consisting of x timeslots (x = 4 in the figure). Each timeslot allows a node

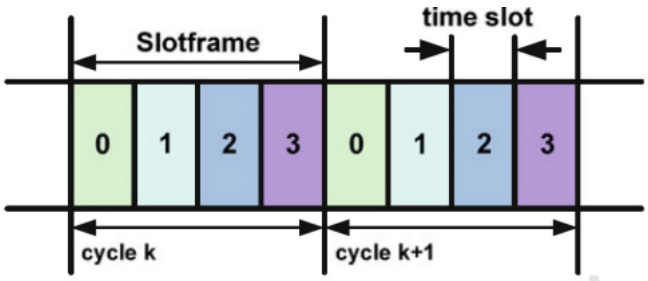


Figure 34: TSCH slotframe and timeslot, with x = 4.

to send max size data and receive ACK. A channel is identifies by the slot associated for bidirectional communication and a frequency offset. TSCH has concurrent transmissions on multiple channels (using different frequencies) and it also blacklist frequencies with poor link quality.

10 Building the IoT

When we develop an IoT system we have a number of devices that are used to create smart objects that are part of a Local Network. The information sensed inside the LN will be provided to a Gateway, that is often the same device through which we send ”inputs” to the smart devices. The Gateway is optional because some devices can directly communicate through

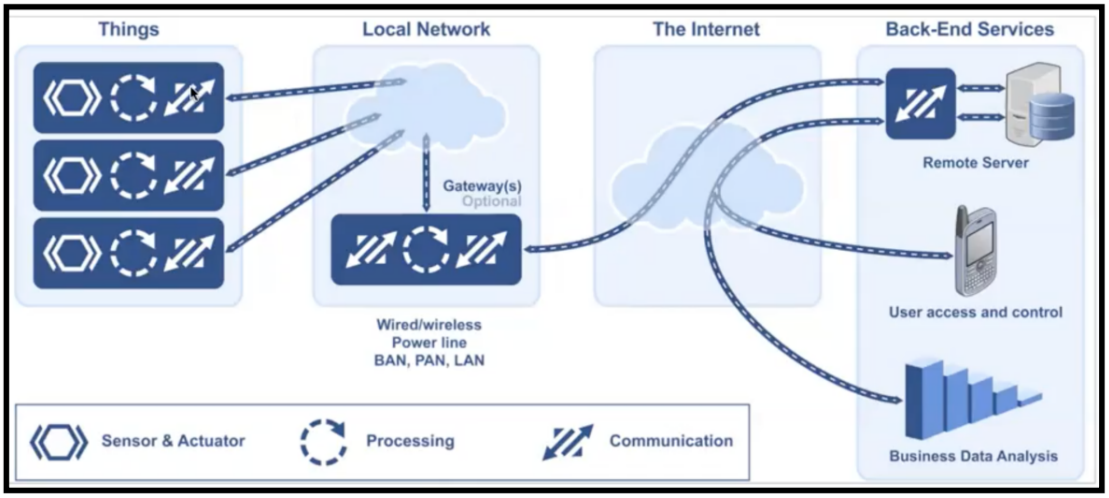


Figure 35: IoT network structure.

the cellular network. Most of the time our gateways/devices communicate from/to a Remote Control Center on a Remote Server. The RCC is an application that runs in cloud.
When we design an IoT system there are some requirements:

- Low power.
- Miniaturization.
- Indoor/outdoor .
- What to sense .
- Data Analysis capability .
- Data rate, range, security support...

10.1 Routing in IoT

We have a set of desirable features:

- Energy aware solutions.
- Scalable to thousands of smart objects.
- Long lasting systems (years of correct functioning).
- Auto-configurating and self-managing systems.
- Robust even in presence of poor links.

10.1.1 Routing Protocol for Low-Power and Lossy Networks - RPL

The approach is proactive and uses distance vector routing. It specifies how to build a destination by building an oriented acyclic graph (DODAG) rooted in the edge router. It supports multi-hop and uses flexible metrics:

- Find paths with the best ETX avoiding non encrypted links, or
- Find the best path in terms of latency avoiding battery operated nodes.

In this way RPL leaves to the administrator the decision on having multiple routing types to carry traffic with different needs.
To build the DODAG the root sends a DIO (DODAG Information Object) message. The neighbours of the root listen to the DIO and decide:

- If a node has neighbours which haven't received the DIO, it selects a parent (according to the chosen metric) and forwards the DIO.
- If a node is a leaf doesn't re-forward anything.

To multicast message in RPL we use a Destination Advertisement Obejct (DAO) message. As a node joins the graph it sends a DAO to its parent which forwards it to the root.
Each node can compute a tuple (destination, node to walk) after the DAO message exchange. We have 2 modes: **storing** and **non-storing**. In the first one a packet goes from a node A until the first common ancestor of A and the destination D; then from the ancestor it goes down to D. In the latter a packet has to be sent up in the tree until it reaches the root that knows how to forward to D. We can see some examples in the image:

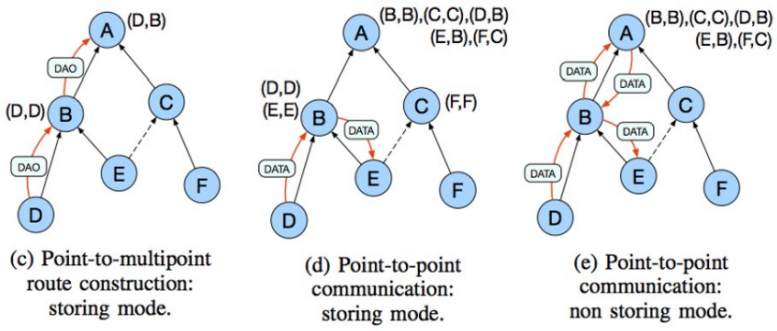


Figure 36: Storing Mode vs. Non-Storing Mode.

DIO messages are sent after a selectable timer or when an inconsistency is detected.

11 Cellular Systems - Beyond GSM

11.1 2G+ - EDGE

Enhanced full rate codec based on Algebraic Code Excited Linear Prediction (ACELP) at 13 Kbit/s. There is a trade-off between voice encoding quality and data rate. The first important step for this trade-off has been done in 2G+ thanks to the Adaptive MultiRate encoding, a new generation codec that adapts to the available channels (half or full rate) and changes the rate depending on the channel propagation conditions.
Another feature is represented by Tandem Free Operation that prefers multiplexing of flows of coded speech signal over transcoding (that degrades voice quality). The multiplexing is done on PCM channels in case of MSs communication.
The improvement of the physical layer brought also enhanced data rates and the possibility to allocate multiple slots to a single MS.
2G+ performs and achieves:

- Triangolarization to localize an MS in a range of 100m from the BTS.
- Number portability (change operator but same number).
- Cordless telephones connecting to Home Base Station.
- SIM application toolkit. No more master-slave relationship between ME and SIM. Now SIM can ask the ME to performs action such as:
  - Set up of a call in a number that the SIM stores.
  - Tone generation.
  - Passing info (from ME to SIM).
  - Execute a certain command.
  - Launch the browser to reach a web address.
- Environments for application/service support, customization, personalization: Mobile Execution Environment; CAMEL.

The radio access technology is **EDGE**:

- Three times more bits per symbol.
- EDGE, thanks to channel encoding, achieves a gross data rate of 59.2 Kbps vs 22,8 Kbps of GSM.
- Carrier bandwdith: 200 kHz.
- Timeslots per frame: 8; frame duration: 4,615 ms.
- radio interface symbol rate: 270ksymbols/s (vs. 270kbits/s in GSM).
- Normal burst: 384 payload bits (116 in GSM).
- Max gross bit rate per time slot: 59,2 kbit/s (22,8 Kbit/s in GSM).

11.1.1 Adaptive Modulation

A dynamic selection of channel coding and modulation based on the Signal Noise Ratio (SNR) of the radio channel. The transmission data rate is reduced or increased based on the quality of the radio channel.

11.1.2 General Packet Radio Service - GPRS

The Internet phenomenon raised the need to support data transmission (not only voice) on cellular systems. The first technology that supported the evolution is GPRS, which uses an IP backbone for packet switching integrated with the circuit switching networks.
This change brought architecture adaptations:

- New type of node, GSN (GPRS Support Node) that are IP routers to support mobility management. We have 2 types:
  - SGSN (or Serving GPRS Support Node) route IP packets from/to a set of MS under their area. The area splitting is more fine grained than before, Location Areas are indeed divided in Routing Areas (RA).
    - Handles user authentication and checks it is entitled to access the service; coordinates encryption.
    - Performs mobility management.
    - Together with BSS radio resource management it reserves radio resources needed to support the requested QoS (Quality of Service).
    - Gathers information useful for billing.
    - Routes information flows from/to the MS.
    - Performs encapsulation and tunneling of packets.
    - Performs logical connections management to/from the MS.
  - GGSN (or Gateway GPRS Support Node) interfaces the cellular network with external packet data networks.
    - It interfaces the cellular operator network with external packet data networks.
    - Performs routing tasks, encapsulation/decapsulation, analyzes and filters arriving messages, and gather info needed for accounting and billing.
    - Stores in its location register the address of the SGSN who are serving the different MS, MS user profiles, and active/standby MS PDP context. Upon request it creates the PDP context: used protocol (e.g., IPv4), MS IP address, requested QoS, ...
  - SGSNs and GGSN are interconnected through an IP backbone.
- At the BSSS level a new functional entity is added denoted Packet Control Unit (PCU) to manage data transmission over the radio links. It deals with dynamic resource allocation between GSM CS and GPRS; and to interconnect MS and SGSN for packet data exchange. Its tasks are:
  - Segmentation and reassembly.
  - Physical channel scheduling.
  - Error detection and management (ACK/NAK, buffering, retransmissions).
  - Access request management and resource allocation.
  - Channel management (power control, congestion management, broadcasting of control messages etc.).

11.1.3 Packet Data Protocol (PDP) Context Activation

An active PDP context related to a user lets the user be visible in the PDN (Packet Data Network) and enables it to send and receive packet. A PDP context activation can be requested after a static IP address is allocated to the MS. A PDP context contains:

- PDP Type (e.g., IPv4).
- Requested QoS class.
- The address of a GGSN that serves as the access point to the external network.

PDP context is stored in all SGSN and GGSN.

11.1.4 Physical and logical channels

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. It supports more information flows (max 8) that are multiplexed over the same channel and it supports transmission of radio blocks. A block is an RCL/MAC block + a Block Check Sequence, 456 bits transmitted in 4 normal bursts in 4 slots in consecutive frames. Besides, this new technology allows the allocation of multiple slots in parallel for the same MS.

Resource allocation

- The network allocates resources as a Temporary Flow Block (TBF) which has associated a Temporary Flow Identity (TFI).
- Before an MS can communicate it has to request a TBF.
- Once PDU have been transmitted the TBF is released.
- The network must allocate resources so that different flows can be multiplexed over the same physical channel.

Control channels

- Packet Common Control Channels - PCCH:
  - Packet Paging Channel - PPCH.
  - Packet Random Access Channel - PRACH.
  - Packet Access Grant Channel - PAGCH.
  - Packet Notification Channel - PNCH. A downlink channel used to notify a group of MS that there are packets for them.
- Packet Broadcast Control Channel - PBCCH.
- Packet Dedicated Control Channels - PDCH:
  - Packet Associated Control Channel - PACCH. Bidirectional channel that transmits info associated to one or more PDTCH. These info are: power control, ACK NAK, re-assignment of resources, assignment of PDCH.
  - Packet Timing Advance Control Channel - PTACCH.

11.1.5 MS States

An MS can be in 3 states:

- **Ready:** Location of an MS known at cell level. Data exchange without paging..
- **Standby:** State entered after some inactive period. After some time, to maintain info on position, the MS pages a user and starts exchange data again going back to Ready.
- **Idle:** After an MS detach or after a timer expires in Standby (no exchange of data in Standy) MS move to Idle. Info on location it's known at LA level.

11.2 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.
There is a hierarchical organization of cells:

- Macrocell.
- Microcell.
- Picocell.

The bandwidth are 1885-2025 Mhz and 2110-220 Mhz, 155 MHz are reserved for terrestrial and 75 MHz for satellite communications. The bandwidth is split in 5 MHz channels that uses Time Division Duplex (TDD) and Frequency Division Duplex (FDD) between uplink and downlink.
UTRAN Functions:

- Controls cell capacity and interference in order to provide an optimal utilization of the wireless interface resources.
- Includes Algorithms for Power Control, Handover, Packet Scheduling, Call Admission Control and Load Control. Among them there are network based functions:
  - **Packet Scheduling:**
    - Controls the UMTS packet access.
    - Handles all non-real time traffic.
    - Decides when a packet transmission is initiated and the bit rate to be used.
  - **Load Control:** ensures system stability and that the network doesn't enter an overload state.
  - **Call Admission Control:** decides whether or not a call is allowed to generate traffic in the network.
- Encryption of the radio channel.
- Congestion control to handle situations of network overload.
- System information broadcasting.
- Micro and macro diversity.

3G uses CDMA (Code Division Multiple Access). A unique identifier (Code) is assigned to each user and all user share same frequency but messages can be decoded using their code, thus identified.

11.3 LTE - 4G

LTE stands for Long Term Evolution. Its requirements are:

- Increased user data rates.
- Uniformity of service provisioning (even at cell edge).
- Improved spectral efficiency.
- Greater flexibility of spectrum usage.
- Reduced delays (connection establishment and transmission latency).
- Low energy consumption at the mobile.
- Seamless mobility and simplified network architecture.
- Able to operate in a wide range of frequency bands and sizes of spectrum allocations (from 1.4 up to 20MHz per carrier).
- Fast connection time (less than 100ms), at least 200 active state users per cell supported by control signaling up to 5MHz and at least 400 users per cell for wider spectrum allocation, one way packet latency = 5ms in light traffic.
- Increased peak rate, uniform performance (requirements on cell edge, performance as 5°percentile of performance).
- Support of mobility up to 350-500Km/h, cells radius 5-100Km.
- Flexible inter-operation with other radio access technologies (service continuity in the migration phase), in particular earlier 3GPP technologies, and non-3GPP technologies (WiFi, CDMA 2000, WiMax).
- Low terminal complexity and power consumption.
- Cost effective deployment due to:
  - One type of node, the BS, named eNodeB.
  - Open interfaces, multi-vendor interoperability.
  - Self optimization and easy management.
  - Packet switching services.
  - Easy deployment and configuration of home base station.

To improve the aggregated data rate we can't operate too much on increasing the bandwidth, this because the radio spectrum is limited and most of the frequencies are already allocated. Thus, LTE improves on the spectral efficiency side.
LTE is based on Orthogonal Frequency-Division Multiplexing (OFDM), in contrast with FDMA/TDMA/U
OFDM was designed on purpose for a packet switching model.

11.3.1 LTE Technologies

It supports multicarrier with OFDMA for downlink and SC-FDMA for uplink. This makes LTE flexible, adaptable and robust and at the same time makes the complexity of the receivers lower.
The technology operates with multiple antennas and it's born to support packet switching. This brought a system architecture evolution:

- Concept of evolved packet system bearer to route IP packets from a gateway of the PDN to the UE.
- The bearer is an IP packet flow with a given QoS between the gateway and the UE. Multiple bearers can be established for an end user providing different QoS. A bearer has a minimum guaranteed bit rate. There exists non GBR bearers. Each bearer has an associated QoS Class Identifier (QCI) and an Allocation and Retention Priority. The QCI is used to determine priority, packet delay budget and acceptable packet loss rate.

11.3.2 Radio Access Network (RAN) Architecture

The high-level architecture of LTE has 3 components:

- The User Equipment.
- The Evolved UMTS Terrestrial Radio Access Network (E-UTRAN).
- The Evolved Packet Core.

The BTS/BSC concept goes away and it's substituted by the enhanced NodeB (eNodeB or eNB). The eNB ensures the necessary QoS for a bearer.
Like in previous standards we have some fundamental units such as:

- **Home Subscriber Server - HSS:** handles user subscription data, PDNs to which user can connect, identity of the MME the user is connected to and authentication center. It is basically HLR + AuC.
- **P-GW:** IP address allocation to UE, QoS enforcement in downlink and inter-working with non 3GPP technologies.
- **S-GW:** transports the IP data traffic between UE and external networks.
- **Mobility Management Entity - MME:** connection set up including paging within a tracking area and security.

11.3.3 LTE Protocol Stack

PDCP performs IP header compression and produces output PDCP-PDU (PDCP Packet Data Unit). The RLC protocol is responsible of the segmenting of the PDU for radio interface transmission. It also performs error correction with the Automatic Repeat Request (ARQ) method. The MAC layer is responsible for scheduling the data according to priorities and multiplexing data to Layer 1 transport blocks. The physical layer performs coding, modulation, antenna and resource mapping.

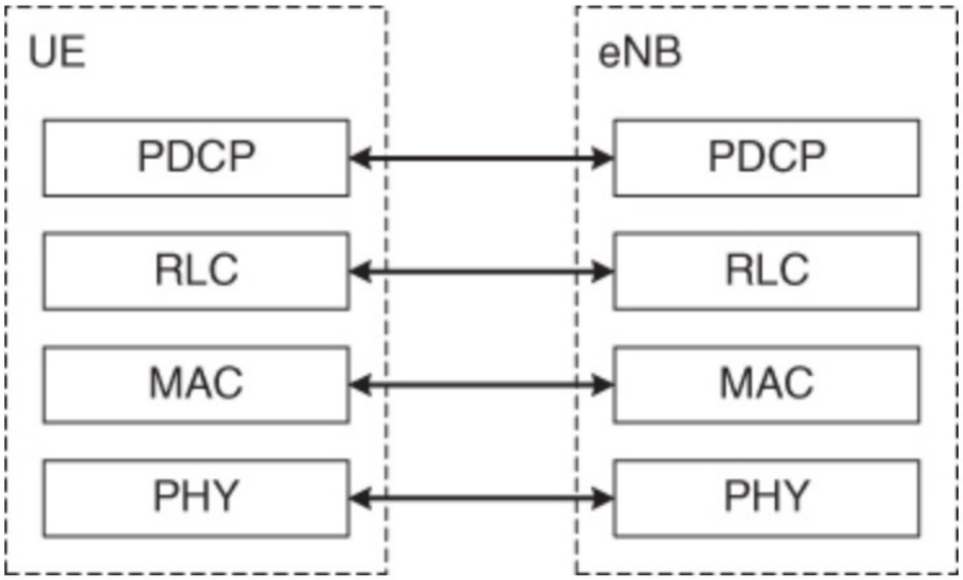


Figure 37: LTE Protocol Stack, UE and e NB.

11.3.4 Scheduling

The scheduler is a unit in the eNodeB and distributes the available radio resources in one cell among the UEs. eNodeB allocates downlink or uplink radio resources based on the downlink data buffered, on Buffer Status Reports (BSRs) received from the UE and based on channel quality indicator reports. eNodeB considers the QoS requirements of each configured radio bearer and can apply two scheduling paradigms:

- **Dynamic scheduling:** assignment of downlink transmission resources and uplink grant messages for the allocation of resources. Valid for specific single subframes, transceive on PDCCH.
- **Persistent scheduling:** resources are semi-statically configured and allocated to a UE for a longer time period.

11.3.5 OFDM

LTE is based on OFDM. The OFDM symbols are grouped into resource blocks that have a total size of 180kHz in the frequency domain and 0.5ms in the time domain. Each user has a number of *resource blocks* in the time frequency grid. The more resource blocks a user gets and the higher the modulation used, the higher the bit-rate. Based on feedback information about the frequency-selective channel conditions from each user, adaptive user-to-subcarrier assignment can be performed, enhancing considerably the total system spectral efficiency.

11.3.6 Physical Layer Adaptation

Modulation scheme and code rate are dynamically selected based on predicted channel conditions. A Channel Quality Indicator provided as feedback by the UEs is used to estimate different channels conditions.

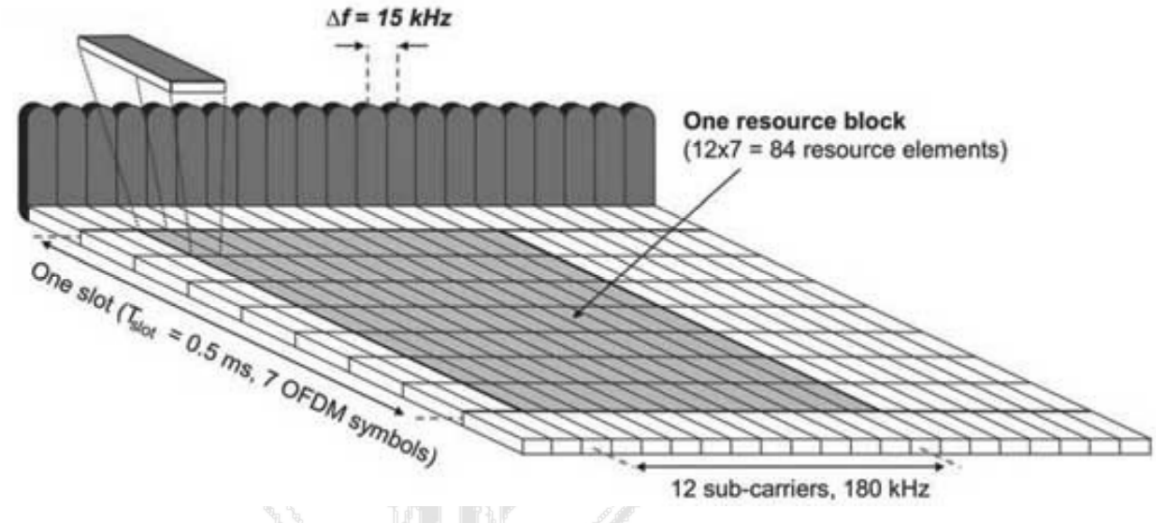


Figure 38: OFDM frequency and time slots.

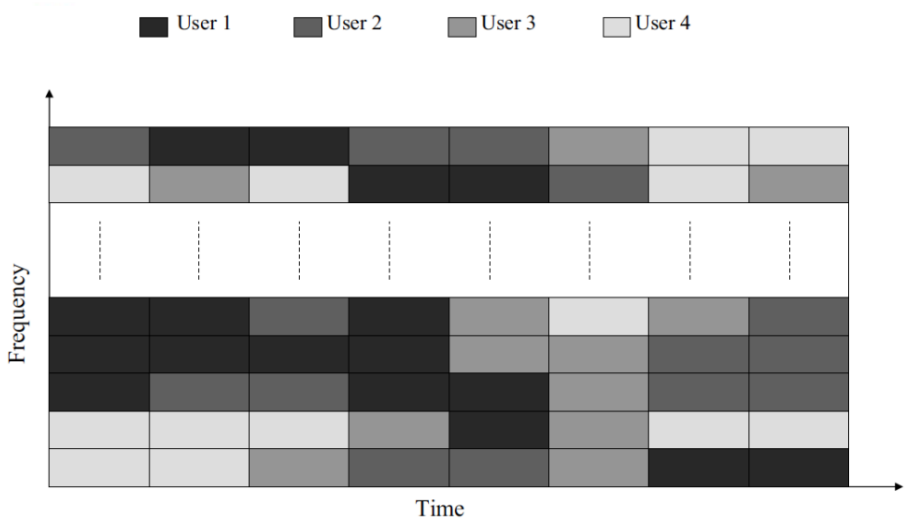


Figure 39: OFDM sample allocation.

11.4 5G

The goal is to provide a delay critical, ultra-reliable, dependable and secure broadband communications service to mobile users. It is not only destined to humans but to billions of smart objects in the emerging IoT network.
In the 5G there is a convergence of different wireless systems. 5G provides 1000x more available throughput in aggregate, as well as 10x more speed to individual end users. The latency is down to 1ms and there is a 90% increase in energy efficiency resulting in a 10 times better battery lifetime for low power devices. The coverage is **really** everywhere every time.
The privacy is assured by design. There are challenges:

- QoS challenge because the variety of services is incredibly high.
- Simplicity challenge, seamless service provided.
- Multi-tenancy challenge: provide services across different infrastructures with different networks coexisting.
- Density challenge (brought by IoT devices).
- Diversity challenge. Must support the increasing diversity of optimized wireless solutions.
- Harnessing challenge: exploit any communication capability.
- Harvesting challenge: exploit energy harvesting to improve lifetime.
- Mobility challenge: seamless mobility across networks/technologies.
- Location and context information challenge: sub-meter localization accuracy.
- Hardening challenge: making communication system robust to attacks and natural disasters.
- Resource management challenge.
- Flexibility challenge: device truly flexible control mechanisms and protocols for relocating functions.
- Identity challenge: provide identity management for any type of device.
- Manageability: improve manageability of networks (reducing human intervention).

The three main paradigms are: flexibility, programmability, openness.

12 IoT Radio Technologies

12.1 Low Power Wide Area Networks - LPWAN

The typical architecture:

- A base station at highly exposed sites serves up to one million sensor nodes.
- Small and cost-efficient sensor nodes communicate using ultra-low power ove long distances. This leads to very weak reception levels.

12.2 LoRa and LoRaWAN

LoRa is an innovative physical layer that provides connectivity to low power smart objects. **LoRaWAN** is a complete stack for building WAN on top of LoRa links. The physical layer is *patented* (one supplier) and the MAC layer standardized (LoRaWAN).

12.2.1 Physical Layer

To reach very far distances LoRa utilizes the *spread spectrum* technique, that consists in transmitting the signal with a frequency band much larger than needed. This technique makes LoRa very robust to interference, multipath and fading.

12.2.2 LoRaWan

It is the communication protocol and architecture that utilizes the LoRa physical layer. There are different three classes of end devices for different application requirements, here in decreasing battery life order and decreasing communication latency:

- **Class A**: each uplink transmission is followed by 2 short downlink receiving windows. They are battery powered sensors:
  - The most energy efficient.
  - Downlink available only after sensor TX.
- **Class B**: like A, but extra receive windows at scheduled times.
  - Energy efficient with latency controlled downlink.
  - Slotted communication synchronized with a beacon (sends schedule).
- **Class C**: continuous receive window, except when transmitting.
  - Devices which can afford to listen continuously.
  - No latency for downlink communication.

12.2.3 Network Server

The centralized network intelligence, responsible for:

- Identifying duplicated packets.
- Data validation and de/multiplexing.
- Localization thanks to time reference for all gateways.

Besides it makes the gateways cheaper because it takes every ”heavy” decision.

12.2.4 Security

There are 2 layers of security:

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

12.2.5 Over The Air Activation - OTAA

An alternative to static activation, it is a join procedure to participate to the network data exchange. When a node wants to join, it will send a request to the network server, with a specific join ID and device ID. The, network server replies with a message that is encrypted with the network key.

12.2.6 Performance Evaluation

LoRa is a pure ALOHA system. There is no synchronization. A node sends a packet immediately if duty cycle is satisfied. In case of collisions there will be a reschedule or cancellation. The maximum throughput is only 18% of the network capacity.

Spreading Factor - SF It is a factor that controls the speed of data transmission. Lower the SF the higher the transmission rate is, but the range of the transmission is lower. SFs are orthogonal, thus signals modulated with different SFs transmitted on the same frequency channel at the same time do not interfere with each other. ,LoRa can sometime also solve interference when two (or more) packets arrive at the same time with ,the same SF. The method used is the "capture effect", i.e. taking only the signal with the highest strength. SFs are not only important for transmission, but also related to the battery life. Higher SFs result in longer active times for the radio transceivers and shorter battery life.

Adaptive Data Rate - ADR LoRa cells can't sustain high loads, thus to increase the network capacity we have to deploy multiple gateways, that can operate on multiple channels at the same time.
LoRa can optimize the SFs allocation thanks to the ADR schema. It is a basic mechanism that says that if we have a certain signal to noise ratio and a given signal strength we can reduce SF, increase the wait and reduce the time on air. Every time we increase the SF we double the time on air. We can think that giving higher SFs (more robust) to far away nodes is better, but actually the random approach results the best.
We may use power control to avoid inter-SF interference but it'll destroy capture opportunities.

12.3 EXPLoRa

It takes into account that only few nodes can work on higher data rates. It balances traffic among SFs and increases capture probability by distributing user working on the same SF in the coverage area. The algorithm works thanks to the **Sequential Waterfilling** that orders users in function of their RSSI and allocates SFs in proportion. In this case the Sequential Waterfilling works dividing the nodes into M sets that see same gateway (nodes are assigned to the closest gateway). Each gateway assigns different SFs keeping the proportions. The interference between cells is proportional on each SF in absence of link constraints.