Internet Of Things' Notes

Daniele Bertagnoli

2022/2023

Contents

1 IoT Technologies

1.1 What is IoT?

The IoT paradigm grounds on three pillars which define the capabilities of the smart things:

- **Identification**, each device must be identifiable.
- **Communication**, each device must be able to communicate.
- **Interaction**, each device must be able to interact with the other devices, the environment, the users and the networks.

A smart thing has its own physical charateristics (size, shape, etc...), has a name and an address with some compute capabilities and communication mechanisms.

IoT is related to entities that interact with the network as producers or consumers of data. We can distinguish between:

- **Sensors**, they measure a physical phenomena and transform it into a digital signal. Typically is powered by a battery and uses an operative system like *Tiny OS*.
- **Actuators**, they receive a digital signal and pereform a physical action based on that signal (turn on a light, play a sound, etc...).
- **Microcontroller**, it manages the sensors and the actuatos. We can also jave some sensors installed on the microcontroller.

The application layer must care about the energy management and try to pereform the operations (sensing and communications) in a way that we can ensure that the devices will be alive long enough.

IoT integrate existing technologies for several usage, such as: healthcare, agricolture and home monitoring. Some of these technologies are re-engineered for IoT purpose:

- **WSN**.
- **M2M**.
- **CPS**.

1.2 Wireless Sensors Networks (WSN)

A WSN is a network composed by sensor nodes. Each node is composed by:

- **Sensors**, we have several *Analogic to Digital Converter (ADC)* that senses a certain physical phenomena.
- **Processor**, it performs a local processing on the sensed data and manage the communication functions.
- **Radio units**, they are used to transmit the data, typically the nodes use short-range communication technologies like Bluetooth and WiFi.
- **Battery**.

In a sensor node we do not have the actuating part. Each node can communicate with a **Master Node (or sink)** that can trasmit all the received data to a remote server through the internet (so it act as **gateway**). The nodes can forward the messages to their neighbors through a multi-path communication in order to reach the master node. One of the challenges is to choose the right routing protocol to forward the messages inside the network.

1.2.1 WSN features

The main features of a WSN are:

- **Fault Tolerence**, if a node fails the WSN remains available.
- **Scalability**, we can increase or decrease the number of nodes without changing the implemantation.
- **Long life time**, we consider:
  - Time at which the first node depletes its battery.
  - Time at which the last node depletes its battery.
  - Percentage of nodes that are alive.
  - Percentage of covered area.
- **Security**, an attack on the WSN can easily compromise the whole network. The management algorithms and protocols of the network extend the attack surface of the system.
- **Programmability**, ensure reuse, robustness and autonomous configuration. The system can self adjust to react to the environment or a failure.

- **Affordability**, an higher number of sensors requires an higher budget.
- **Heterogenity**, supports a wide number of sensors.
- **Mobility**, some sensors can move around, so it must adapt to an ever changing topology.

1.2.2 Energy Management

When we are deploying a WSN, we must care about the energy management. We can apply different techniques to increase the network life time:

- **Duty cycle**, some sensors can be turned off in some times.
- **Adjusting the sensing range**, we can reduce the sensing range by avoiding redundant sensed areas. So we can shrink the signals from different sensors and ensure that each sensor cover a different area.
- **Selective activation**, we can alternate the role of the sensors and let other sensors sleep.

The duty cycle and the selective activation requires that the routing protocol must be able to manage the possibility of that one sensors can be in the spleeping phase.

1.2.3 WSN Stack

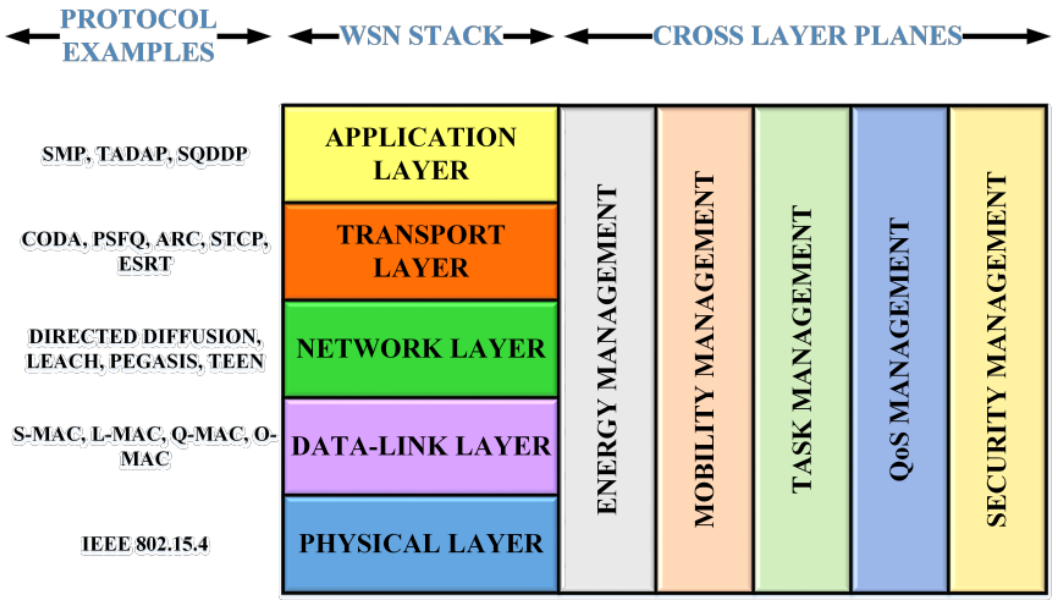
Is composed by 5 layers:

1. **Application**, is responsible for traffic management and software interfaces.
2. **Transport**, ensures reliability and congestion control of the packets arriving or leaving. Includes protocol based mechanism for loss recognition and loss recovery.
3. **Network**, manages the routing of packets and the routing issues (*Dead end*; *Sink-hole* all the devices near the master node have depleted their battery since all the communications are forwarded through them; *Cross-layering* for energy, movement and task management).
4. **Data Link**, is responsible for the medium access control (MAC) functions such as multiplexing and demultiplexing.
5. **Physical**, enables transmission of signals over a physical medium between nodes. In WSN the medium is strictly wireless (with low data rate and low energy consumption).

There are also 5 cross layer planes:

1. **Energy Management**.
2. **Mobility Management**.
3. **Task Management**, task defines the amount of data that must be processed.
4. **Quality of Service (QoS) Management**, it highly affects the communication service.
5. **Security Management**

Each of these planes have consequences on all the stack layers.



1.2.4 WSN Types

There are several tupetypes of WSN:

- **Wireless Multimedia Sensor Networks**, composed by *Vectorial sensors* that capture images and sound. Expensive sensors.
- **Wireless Underwater Sensor Networks**, it has several challenges due to the environment. The mobility and the communications are more difficult considering the underwater scenario.
- **Wirelss Underground Sensot Networks**, like for the underwater sensor network the main challenge is to manage all the problems caused from the environment.

- **Wireless Mobile Sensor Networks**, these networks relies on an high mobility (smartphone, laptops or GPS) where the signal must reach the node considering that the deivce is moving around the area.

1.3 Machine to Machine (M2M)

M2M paradigm **provides communication between two or more machines/devices without any human intervention** (some examples: ATM machines, vending machines alerting a remote inventory of the need to refill). As the other sensor network systems (IoT, WSN, CPS), the aim is to sense a physical phenomena, produce some data and eventually use them through the actuators. M2M is also known as **Machine Type Communication (MTC)**, the main charateristics are:

- **Heterogeneous markets**.
- **Low data footprint**, small sized data.
- **Low cost mantainance and integration**.
- **Large networks without human intervention**.

M2M devices are generally **static devices** and the data transmission between the machines are **highly time-bounded**, but not real time and **delay tolerant**. The communications are organized in tiny packets (due to the tiny data footprint). Typically there are no actuators, the main goal of the machine is to monitor an event, so there is a low energy consume. These networks are easy to be extended with other devices and are always connected with the network and the cloud. M2M paradigm can be described with:

- **M2M Networking Model**, focuses on the networking components. The communication type is **wireless**, the devices can be connected with the network through a gateway or directly. Each device manage the routing in an autonomous way. We can categorize the devices in:
  - **Low-end devices**, cheap and low capabilities of auto-configuration, power saving and data aggregation. They are static devices so they need an highly dense deployment. Used for environmental monitoring applications. Produces Non IP-based data.
  - **Mid-end devices**, more expensive than low-end and may have mobility. They have localization, intelligence, networking management and support for QoS.
  - **High-end devices**, these devices require low-density of deployment. They are mobile deivces that can handle multimedia data (such as video or audio).

The **area network** (also known as *device domain*) includes multiple M2M devices that communicates with eachother. The **gateway** provides the communcations between the M2M devices and the Internet. It also ensure that the devices can be accessed from the outside of the network. The **communication network** (also known as *network domain*) consists of the technologies of communication between M2M devices, the gateway and various applications. These networks can be:

  - **IP-based**, M2M devices produce packets forwarded to the internet through the gateway.
  - **Non IP-based**, the M2M devies produce data that are sent to the gateway, then the gateway will packetize those data and forward them to the internet.

- **M2M Service Ecosystem**, classifies the M2M devices based on the **needs of service of the M2M platform**. The ecosystem can be divided in 4 domanins:
  - **M2M Area Networks**, same of the networking model.
  - **Core Network**, form the communication infrastructure of the M2M, can be both wired or wireless.
  - **M2M Service Platform**, is divided in:
    1. **Device Management Platform**, enables anytime and anywhere access between internet platforms and the registered objects. Each object has a database that can be accessed by the end-users.
    2. **User Management Platform**, manages the user profiles, their registration and modification to the platform.
    3. **Data and Analytics Platform**, provides services based on the data collected by the devices. It also merges the heterogeneous data by processing them for management purposes.
    4. **User Acces Platform**, user can access the platform via applications or web that will redirect the user to the *service providers* that keep track of the registered devices and users.
  - **Stakeholders**, we can distinguish between 5 different kind of stakeholders:
    * **Device Providers**.
    * **Internet Service Providers (ISP)**.
    * **Platform Providers**.
    * **Service Providers**.
    * **Service Users**.

1.4 Cyber Physical Systems (CPS)

CPS are networked monitoring and controlling systems **governed by feedback-based algorithms**. The difference between CPS and WSN or M2M paradigms is the inclusion on inclusion of feedback systems. The sensors will sense a phenomena, based on that the actuators will perform an action. After that a feedback is sent to the actuators that will choose the next action until the final goal is reached. The main features are **real-timeliness**, so the system depends on the the real-time communication and feedback to provide control over the environment. These systems are **intelligent**, so they can perform adaptive decisions according to the feedback, this requires coordination between the devices. Obviously, this implies also the ability to predict the output of the action to avoid harmfull actions. CPS are composed by a vast amount of **heterogeneous** devices, so the software must be able to manage them in the right way. CPS are scalable systems in terms of network bandwidth, number of sensors and actuators, size etc... **Secure** these systems is crucial to avoid harmfull situations.

1.4.1 Digital Twins

Digital Twins are behevioral and functional mathematical models of the physical system. They are a digital copy of the systems, created to study and replicate the real system in order to save money and perform experiments to test the whole systems.

2 Emergency of IoT

The total number of globally connected devices is ≈ 50 bilions, and is still rising up. This means that the **IoT systems must be efficient, scalable and able to manage intermittent and unstable connections**. M2M, CPS and WSN are simpler than IoT systems, infact, these systems are included inside the IoT system definition. IoT systems also includes the **Web of System (WoS)** that provides the access to the resources through the web interfaces.

2.1 IoT Paradigm

We divide the IoT paradigm into four planes:

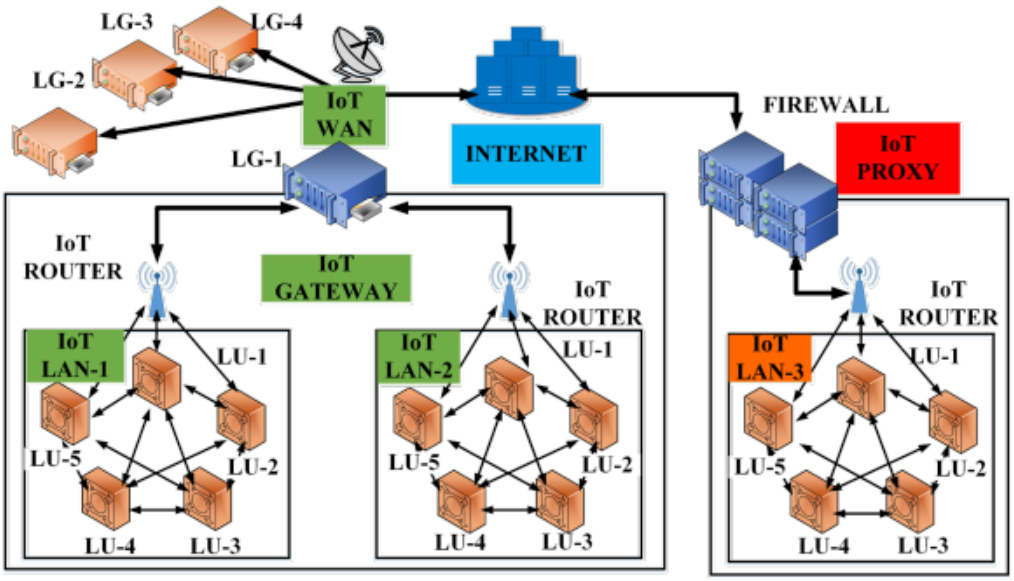
- **Service**, this plane is composed by:
  - **Things of Devices**.
  - **Low-power Connectivity**.
- **Local Connectivity**.

- **Global Connectivity**.
- **Processing**.

2.1.1 Networking Components

An IoT implemantation includes several components from the networking view:

- **IoT Node**, they are typically composed by a *sensor*, *processor* and *radio antenna* used to communicate with the network. Each node has an **Local Unique Device Identifier (LU-x)** that are unique only in the LAN.
- **IoT Router**, it manages the routing of the packets and keeps the traffic correctly balanced.
- **IoT LAN (Local Area Network)**, enables local connectivity and typically consists of short-range connectivity technologies.
- **IoT WAN (Wide Area Network)**, connects various local networks.
- **IoT Gateway**, is a router which connects LAN to WAN.
- **IoT Proxy**, is a device which is part of the application layer and provides security functionalities.



In the network layer both IPv4 (32 bit addresses) and IPv6 (128 addresses) are used to address a machine. The mapping of a device's logical address and its physical address is performed by a mechanism called **Address Resolution Address (ARP)**. The IPv6 addresses can be divided in 7 types:

- **Global Unicast (GUA)**, unique addresses that can be assigned typically to gateways, proxies or WANs.
- **Multicast**, these addresses provides tansmissions from a single device to multiple destinations.
- **Link Local (LL)**, are valid in LAN only, so can be repeat in other LANs.
- **Unique Local (ULA)**, similar to LL but are unique in the organization scope. the internet, typically used for organizations.
- **Loopback**, used from the developers and testers for checks.
- **Unspecified**, all the bit of the address are set to 0.
- **Solicited-node Muticast**, multicast address based on the IPv6 address of a node.

We can distinguish between **7 routing strategies**: (DO NOT REMEMBER)

1. **Class-1**, the nodes are connected only with themselfs, there are no gateways. LL addresses are used to identify the nodes. The communication can be direct or through other nodes.
2. **Class-2**, we have more than one Class-1 networks (the nodes inside can communicate with eachother by using the LL addresses) that communicate by using a gateway. To identify the nodes we use ULA and GUA addresses.
3. **Class-3**, we have one LAN connected with a gateway that uses only ULA to identify the nodes.
4. **Class-4**, we have a LAN connected with a proxy that act as a gateway between the LAN and Internet.
5. **Class-5**, same as Class-4 but the nodes cannot communicate directly with eachother.
6. **Class-6**, same as Class-5 but all the nodes are addressed by using GUA.
7. **Class-7**, same as Class-5 but there are multiple gateways.

2.1.2 Tunneling

Is a networking technique in which data from private network can be streamed through a public network as encapsulated packets.

2.1.3 Multihoming

Is a network configuration in which the nodes are connected to multiple networks simultaneously.

2.1.4 Addressing Mobile Devices

One of the main challenges with IoT networks is **addressing considering mobility**. We can use 3 techniques to manage the movement of a device from a LAN to another LAN:

- **Global Prefix Change**, suppose that each LAN has a prefix and uses local address to identify their own nodes. If the device moves from a LAN A to a LAN B, can happen that its local address was already used in the LAN B. To avoid this, it is prudent to avoid static IP addresses.
-
- **Remote Anchoring**, this technique is hard to implement. Each node has a global address that is not affected from LAN changes. This address is provided from an *anchor point* which use the tunneling to communicate the address.

TODO chiedi chiarimenti.

2.2 Sensors, Actuators and Transducers

The main goal of most of the IoT applications is to sense a phenomena and then actuate an action based on the sensed data. Both of sensing and actuating actions are based on the **transduction**, so the energy conversion from one form to another pereformed from a *transducer*.

2.2.1 Sensors

A sensor is a device which can **measure and quantify the ambient changes**. They generate responses to an external phenomenon by converting them into electrical signals. A sensor is sensible only to the measured property and does not influence that property. Based on the sensor type and the sensing activity, sensors have different power requirements, typically these devices have low-power requirements. Sensors is composed by **sensing unit, processing unit, radio and battery**, they rely on wireless communication. Each sensor has three main properties:

1. **Resolution**, indicates the smallest change that a sensors can detect.
2. **Accuracy**, is the ability to measure the environment as close to its true measure.
3. **Precision**, is the property related to the repeatability of the sensor's measurements. The same measurement repeated multiple times should give the same results.

The **sensing range** of a sensor node defines the area in which the sensor's measurements can be considered as reliable. One of the typical approach to optimize the sensing range of a network is the **K-coverage**, where K means that each point of the sensed area is sensed simultaneously by at least K sensors. This technique implies redundancy but is usefull for the neftwork to be more fault tolerant (if a sensors dies, we still have K-1 sensors that sense a certain point). Sensors can be divided into two groups based on the power requirements:

- **Active Sensors**, they emit energy in order to detect the eventual reflected energy:
  - **RADAR** (Radio Detection and Ranging), uses *radio waves* to detect and locate objects based on time taken for the waves to return. Other paramters as intensity, frequency and phase can be used to gather additional information.
  - **LIDAR** (Light Detection and Ranging), uses *laser light* to measure distances and create high detailed 3D maps based on the time taken for the light to return. Other paramters as intensity and spectrum can be used to gather additional information about the object's materials.
  - **SONAR** (Sound Detection and Ranging), uses *sound waves* to map underwater environments or for surveillance applications. Some waves frequencies can also penetrate materials such as water and mud.
- **Passive Sensors**, they do not emit energy but instead they detect the natural emitted or reflected energy by the objects.

Sensors can be also categorized based on the output:

- **Analog Sensors**, generates output singals or voltage which are proportional to the measured quantity and are also continuous in time and amplitude. The conversion to the digital signal is perfomed by and ADC.
- **Digital Sensors**, these sensors generate the output of discrete-time digital representation (time, amplitude or both) of a quantity being measured. Typically the output is binary 1 or 0 associated to ON and OFF.

Another classification that can be done over the sensors is based on the **measured property type**:

- **Scalar Sensors**, generates output proportional to the *magnitude* of the measured quantity. Scalar physical quantities are those where only the magnitude of the signal is sufficient for describing the phenomenon (e.g. temperature, pressure, color, etc...).
- **Vector Sensors**, these sensors are affected by the magnitude as well but also from *direction and/or orientation* of the property (e.g. velocity).

2.2.2 Actuators

An actuator is a device which performs physically heavier tasks (such as moving or changing orientation of objects) when required from the system. The control system of an actuator can be either **mechanical system or electronic system or software-based system**. The main charateristics of these devices are:

- **Weight**, the physical weight of actuators can be a limitation in some applications.
- **Power Rating**, defies the minimum and the maximum operating power an actuator can safely withstand without damage to itself.
- **Torque to Weight Ratio**.
- **Stiffness**, the resistance against material deformation.

2.3 IoT Processing Topologies

The sensor systems generate a huge heterogeneous amount of data with also different rates. We can distinguish between two main types of data based on how the can be accessed:

- **Structured Data**, text data with a pre-defined structure, they are associated with *relational database management systems*.
- **Unstructured Data**, data without a pre-defined structure.

Due to the vast amout of types of data, is important to process them with techniques. In the IoT networks is important to decide when and where the system has to process these data based on the application. Application can be

- **Very Time Critical**
- **Time Critical**
- **Normal**

So based on these categories we must choose when the data should be processed. Data can be processed:

- **On-site**, data are processed by the source itself. This is an expensive solution for the real-time applications which are not delay tolerant.
- **Off-site**, data are sent to an external server that will perform the required manipulations and analysis. This is a cheap solution for those applications that have some delay tolerence. The **collaborative topology** allows the sensors to creating a mesh for the data processing. This architecture can also use **offloading techniques** in order to distribute the computation among the network based on:
  - **Data generation rate**
  - **Bandwidth**
  - **Critical applications**

  The offloading can be perfomed in several architecture's areas:
  - **Edge**, data are processed near the source by local *network clusters*.
  - **Fog**, data are processed by a *decentralized computing infrastructure* in order to conserve bandwidth and keep the latency low.
  - **Cloud**, cloud computing means use distributed system through the internet, this guarantees high scalability but with high latency.

2.4 How to choose the devices

When we have to deploy a network for an application, we must do some consideration about the devices that we will deploy in order to develop the right sensing solution:

1. **Size**, the size is crucial for some applications. Moreover, larger dimensions imply a larger energy consumption by the hardware.
2. **Energy**, the energy requirements are responsible for the life-time of the network. A greater energy consumption means also that the batteries must be replaced more frequently.
3. **Cost**, a cheaper cost of a device enables a much higher density of deployment.
4. **Memory (volatile and non-volatile)**, determine the capabilities of the device. Higher amount of memory implies a cost increment.

5. **Processing Power**, determines the processing features of the device and in which applications it can be used.

6. **I/O Ratio**.

2.5 IoT Networking Protocols

2.5.1 Wireless Channel

The **wireless channel is shared** so must use some protocols as *CSMA/CA* to prevent collision over the medium. We cannot use *collision detection* on wireless networks since the wireless nodes are not able to send a signal and listen the channel simultaneously. Based on the technology used (and the protocol) we have different **data transfer rates**.

The wireless technology uses electromagnetic waves, they can be described by using:

- **Amplitude**, how strong is the signal and is proportional to transmission energy. When two signals interfer with eachother, the amplitudes are summed.
- **Frequency**, the wave's tone. The frequency also affects the wave length an is the portion of wireless spectrum in which we are transmitting. Based on the wave length we decide the size of the antenna that we should use. Typically:
  - High frequencies are good for short communications and are affected by obstacles.
  - Low frequencies are good for long communcations and are less affected by obstacles.
- **Phase**, shift of the wave respect to another signal. A *positive phase* means that the wave has a left-shift, a *negative phase* means that the wave has a right-shift respect to the main signal.

Based on the phase and amplitude, the signal can be attenuated or amplificated from the other signals. We can use **amplifiers** to prevent the signal decay, this device will increase the amplitude of a signal.

The wireless signal can be affected by:

- **Fading**, the **wireless signal attenuates by increasing the distance**, in particular, the power is dissipated in every direction (sphere propagation). The signal will decrease with an inverse ratio respect to the distance (**long term fading**), but we can observe fading even if we are not moving due to fluctuations of the signal (**short term fading**??).
- **Shadowing**, the signal cannot be received since the receiver is behind and obstacle.
- **Reflection**, the signal is reflected from an object.

- **Refraction**, the signal pass through a medium (water, air, etc...) but its direction is modified.
- **Scattering**, the signal is reflected in multple direcitons.
- **Difraction**, the signal is refracted in multiple directions.

Due to these events, the signal follows a **multipath propagation**, so it can reach the destination by following several paths. This can cause interfaces since it can be delivered in different phases. The **SNR (Signal to Noise Ratio)** indicates the quality of the signal respect to the noise. By increasing the power we can increase the SNR, but this will increase also the noise perceived from the other transmitters. The **BER (Bit Error Rate)** indicates the possibility of bit flippings when the signal is received.

Hidden Terminal Problem: We have node A, B and C, where A and C cannot hear eachother and both can communicate with B. Since they are not able to detect communcations, collisions can occur on B.

Exposed Terminal Problem: We have node A, B, C and D where only C can communicate with D. C detects the communication between A and B and does not communicate with D to avoid collisions.

We can distinguish between 3 ranges where the signal can be received and that depends also from the receiver sensitivity:

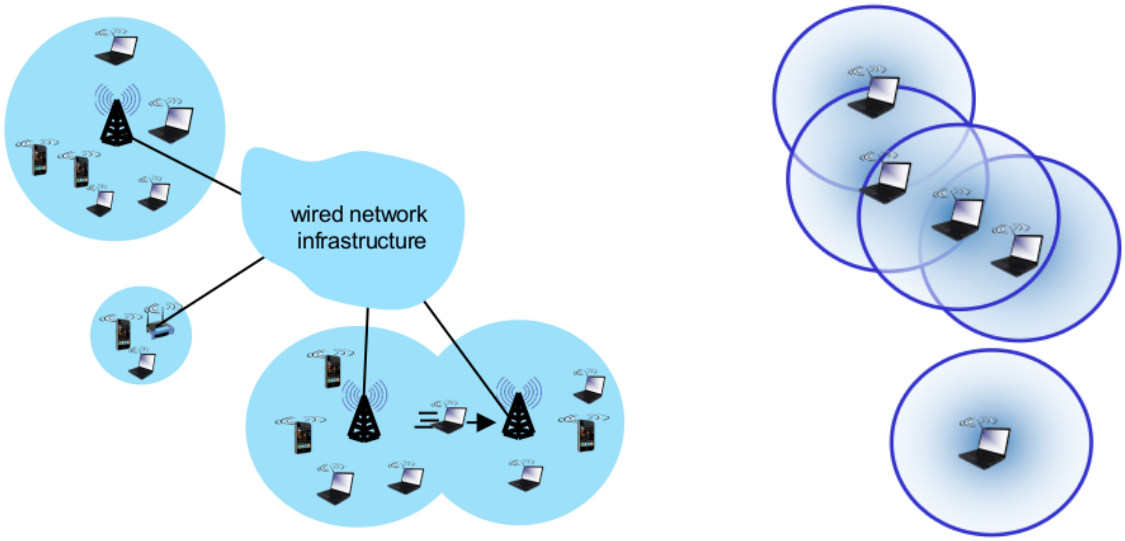
1. **Transmission Range**, the signal can be translated with a low error rate.
2. **Detection Range**, the signal is detected but the communcation is not possible.
3. **Interference Range**, signal may not be detected and becomes part of the background noise.

We can calculate the **outage probability** is the probability that the received power is lower than a certain threshold.

2.5.2 Network Architecture

IoT typically relies on wireless technology to exchange information between nodes in the network. In a network we have **wireless hosts**, network nodes, that communicate with eachother. There are also **base stations** which are devices connected through a wired connection to a **wired network infrastructure**. Base stations are responsible for forwarding the packets between the wired and wireless network. Thanks to the **wireless link** the base station can communicate with the wireless nodes. Since the nodes are wireless, is also possible that they are mobile and so they can switch

between two access point, this is called **handoff**. Each access point has its own coverage area in which it will operate. It can happen that nodes forwards their packets through a multi-hop path to reach the gateway and share the packets outside of the network. We can also have **ad-hoc networks** where there are no wired infrastructure or base stations, the wireless devices organize themselves into a network.



(a) Infrastructured Network

(b) Infrastructure-less Network

2.5.3 IEEE 802.11 Wireless LAN

Is the WiFi protocol used for the communications. The architecture is composed by **Basic Service Sets (BSS)** connected by a **router/switch** to the internet. Each BSS is composed by **wirelss hosts and access points (AP)** (base stations). The shared medium is divided into **channels at different frequencies**. Each AP uses a channel to communicate with the hosts in the communicating area, however, collisions are still possible since the neighboring AP can use the same channel. When an host wants to join a new BSS, it must associates with the AP:

1. The host scans the channels looking for the **beacon frames** which contains the AP's information. This can be done in two ways from the host's point of view:
   - **Passive Scanning**, the AP broadcasts the beacon frames over the selected channel.
   - **Active Scanning**, the host broadcasts the association request and the AP will reply with the beacon frame.

2. The host selects the AP and eventually perform an authentication.

3. The host runs the DHCP in order to obtain an IP by the AP.

IEEE uses a MAC protocol combined with CSMA/CA in order to avoid collisions:

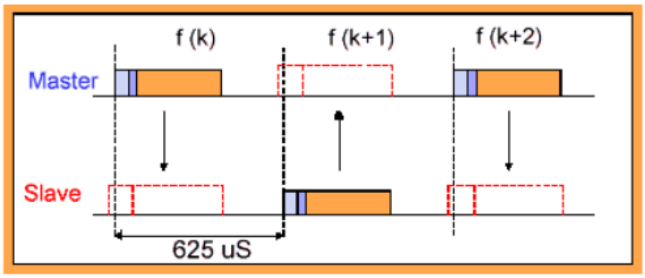
1. Sender listens the medium for a DIFS, if the channel is idle then it will send the packet, otherwise it will wait for a backoff time. When the backoff time is expired and no ACK has been received, the backoff is incresed.

2. When the receiver has received the packet, it will forward an ACK.

The sender reserves the channel by using **RTS packets**. When the receiver has received an RTS it will send back **CTS packets** that are received from every node that can hear. In this way all the neighbors know that that node is busy until an ACK is received from it. In order to save power, the nodes can decide to sleep until the next beacon frame. In this way the AP knows that it cannot comminicate with them for a cycle. All the packets that should be delivered to that nodes are sent only when they will be available.

2.5.4 Bluetooth

Bluetooth is a **short-range wireless technology** with low-cost and low-power usage. This technology is free and worldwide opened but the communcation can be attenuated due to the huge amount of different waves (microwaves, phone waves, etc...). **Bluetooth is based on the creation of small ad-hoc master-slave private networks with an automatic synchronization** (data are kept synchronized without using a cable). We can also connect to internet through the **bluetooth access points** that act as gateways. Each network is called **piconet** and is composed by a master which controls up to 7 slaves. In particular, the master polls a slave that can reply to that request. In order to communicate both **Time Division Multiplexing (TDM)** and **Frequency Division Multiplexing** are used (so each communication step uses a time slot and a frequency slot). **The master knows the schedule of each slave**, in fact, they can go to sleep in order to save battery or to work in an another piconet. These small networks are self-assembled and when a new piconet is created, the master sends the **frequency hopping schedule** to all the slaves. At each communcation step, the frequency is switched to a new one (total of 79 1MHz differnt frequencies) to reduce the collision probability (1600 hops per second). Since at most 8 devices can be part of a piconet, if we want to communicate with more than 8 devices, we can create a **scatternet**. These network are composed by multiple piconets which share a node called **bridge**. There is no synchronization between piconets of a scatternet.



When a device wants to communicate with another device, it will start an **inquiry procedure** to discover the neighboring devices (within 10 meters in 10.24 seconds). Since the two devices are not connected, they probably are not transmitting/receiving on the same frequency. The sender will forward a request over 16 channels called **inquiry train** for 128 times, the receiver will scan the 16 channels until it receives a connection request and replies with an **inquiry response**. These response are **FHS packets** that are sent after a random time between 0 and 0.32 seconds, this to avoid collisions. The **FHS packets are composed by the device ID and a clock**. The clock is used by the master in order to understand when the slave can be polled. A this point, the master sends a **page message** (specific messages) to the selected device. The slave is in the **page scan state** and it is waiting for a message. When the page message is received, the slave enters in **page response state** and will send to the master the response. Finally the master sends back to the slave the frequency hopping scheme and the slave communicates to the master if it an active member of the piconet. Since the channels can be noisy, to ensure that the trains are not lost, these 128 trains are sent every 10 seconds and a device can reply multiple times to ensure that the responses are delivered. Moreover, each device implements a cyclic sleeping phase, so it will perform the scan enough frequently to guarantee that it will be up when the channel train is sent.

Bluetooth uses 4 protocols which compose the protocol stack:

- **Core Protocol Group**, composed by several layers:
  - **Radio Frequency Layer(RF)**, it sends the bit streams by using signal modulation techniques. The first versions of bluetooth were using *frequency shifting*, so change the frequency of the signal based on the bit that we want to transmit. The newest versions use *phase shifting*, so if the wave starts with the minimum is a 0, if the wave starts with the maximum is a 1. This new technique allows a 3Mbps data rate (instead of 1Mbps). Based on the transmission range used, we can distinguish 3 classes:
    1. Class 1, up to 100 meters.
    2. Class 2, up to 10 meters.
    3. Class 3, up to 1 meter.

- **Baseband Layer**, it provides the device discovery mechanism and the frequency hop sequencies for synchronization and transmissions. It also establishes two types of physcal links:
  - **Synchronous Connection Oriented (SCO)**, for timebounded communications (e.g. voice transmission). There is no retransmission (since is real-time communication). This type of channel will continue to work even if the device is in hold mode.
  - **Asynchronous Connection-Less (ACL)**, for sporadic data transmissions. The data integrity is checked through error checking and retransmission.

  This layer also handles the encryption keys generation.
- **Link Manager**, it performs all the link creation, management and termination operation. In particular it controls the **operation modes** of the devices (when and if the device goes in sleep mode) and manages the ACL and SCO links. It provides also 4 **connection modes**:
  - **Active**, is the mode in which the nodes can communicate with the master (at most 7 active nodes in a piconet). In this mode, the node can be polled by the master in any communication frame.
  - **Hold**, is a low-power modality where the node listen less (ACL are not supported but SCO can still work). The node is still part of the piconet and after a certain time it will come back in active node.
  - **Sniff**, is a low-power modality, very similar to hold mode. In this modality the node's activities are reduced for a fixed amount of time that is repeated for a ficed number of times. The master can poll the slave between a *sniff time slot* and the next one.
  - **Park**, the node is sleeping so the master can let another node join the piconet. We can have up to 255 parked nodes. A parked node will wake up periodically in order to listen for broadcast messages sent by the master.

  Link manager also provides security mechanisms for choosing the encryption mode and coordinating the encryption keys.
- **Logical Link Control And Adaptation Protocol (L2CAP)**, it performs 4 main functions:
  - **Creation and termination of logical links**.
  - **Enforcing and defining the QoS requirements**, the applications can require specific QoS (e.g. latency, delay variation, etc...), this layer tries to satisfy those requirements and eventually notifies the applications if the link is not able to do that.

    - **Adapting data through the Segmentation and Reassembly**, Baseband layer can manage smaller packets than the L2CAP layer. So the packets destinated to the baseband layer are segmentated, the packets that comes from the baseband to the L2CAP are reassemblated.
    - **Perform multiplexing to support multiple connections**, applications can access L2CAP by using different protocols:
        - RFCOMM, is a cable replcement protocol which allows the applications to interface themselves with emulated serial ports.
        - Service Discovery Protocol (SDP), is based on a client/server model, the SDP server is a bluetooth device which offers a service (e.g. printers), the SDP client is a bluetooth device that can use the service. The client can query the server to retrive the available services.

- **Cable Replacement Protocol**, it includes the RFCOMM protocol.
- **Adopted Protocol**, protocols adopted from standard models, some examples are Point-to-Point Protocol (PPP), Internet Protocol (IP), Wireless Application Protocol (WAP).
- **AT Commands**, command stes used to specify bluetooth parameters.

The Baseband layer and the link manager are implemented on the chipset, the other components are implemented on the host. The **Host Controller Interface (HCI)** will manage the communcation between the physical layers and the higher layers. This interfaces handles the traffic between the two parts since the may produce/send data at different rates.

A variant of the Bluetooth technology is the **Bluetooth Low Energy (BLE)**, it works in the same way as the classic bluetooth, but uses 40 channels of 2MHz instead of 79 channels of 1MHz. This is done to reduce the energy consumption.

2.5.5 Low Power Wide Area Communications (LPWAN) and LoRa

It is a **long range communcation technology which uses low power** to send data. The main drawback is related to the **low datarate** achieved with this communcation technique. Due to the low datarate, LPWANs are suitable for environmental monitoring applications where the need is to send only few Kbps. The main component is the **base station** (typically a very tall tower) that can serve up to one milion sensor simultaneously. LPWANs can be divided into **three categories**:

- **Network based on cellurar infrastructure**
- **Network based third-party infrastructure**

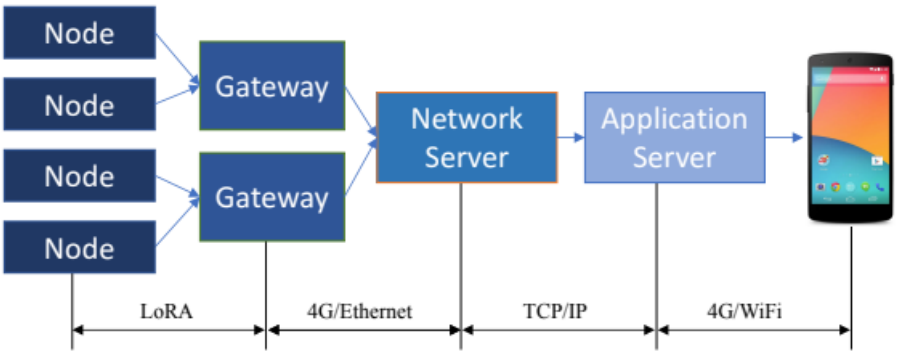
- **Autonomous LPWAN Networks without third-party infrastructure**

One of the main challenges is related to the very weak low transmission power, all the **produced signals are very weak**, thus making more **difficult to distinct them from the background noise**. Moreover, the base stations are highly exposed to other signals and will also complicate the signal reception. We cannot use CSMA beacause of the hidden node problem, so we need to spread the signals over a larger transmission spectrum to make collisions less probables. The most popular protocols which implements this technology are:

- **LoRa**
- **SigFox**
- **NB-IoT**

We will focus on LoRa, it transmits over license-free radio frequencies (433 MHz, 868 MHz, etc...), but the physical layer that LoRa uses is **patented**. Is deployed in **star or star-of-stars topologies**. The base station can communicate with the nodes through the wireless channel using the LoRa physical layer. The base station acts as **gateway** through a **backbone IP-based network** and gather data from the sensors to the network server by converting RF packets to IP packets. The gateways are always connected to the power and act as transparent bridge. There is **no association between the nodes and the base stations**, so they can even broadcast messages to multiple base stations. The first duty of the network server is to **drop the duplicated messages**, so node's broadcasted messages or messages forwarded from different nodes but related to the same phenomenon. This can be done through the use of **Voronoi diagrams**, a geometrical construction over the area of interest to assign a partition of the area to each sensor. The sensor responsible for the partition is the one which best monitors that portion of the area. Each geographic point in the **responsability region** of a sensor is assigned to that sensor if it is the closest sensor to that point. To design the edge of the partitions, we can draw the line between two sensors and then consider the **bisector** as an edge. The Voronoi diagram can be:

- Done in a decentralized manner, each node will create its own responsability area, ignoring all the other nodes that are to far way from it.
- Can be managed by the network server to avoid inconsistencies that can occur with the distributed solution. This can happen when the nodes cannot hear eachother thus making impossible to build the right diagram.



In order to reach very far distances, LoRa use the **spread spectrum**. By variating the **chirp-spread spectrum** the signal is more robust to interferences, fading and multipath. An important concept is the **spreading factor**, it defines the slope of the **linear function through the frequency is modulated by**. By using **different spreading factors we can transmit multiple signals at the same time** since we are able to distinguish the signals based on the spreading factor. The device can choose among 6 SF, the smaller is the SF the more is the bitrate achieved, this beacause we have a lower **time-on-air** (time used by the frequency to reach the peak thus representing a bit). Obviously, depending on the SF also the distance covered by the signal is different, the less is the SF the more is the coverage.

The **LoRaWAN is a communication protocol and architecture which uses the LoRa physical layer**. It supports the bi-directional communication, mobility and localization.

The **LoRaWAN security** is both for the network and application security. Network security ensures the authenticity of the nodes in the network, for the application layer security ensures the network operator cannot access the end user's data. In LoRaWAN we have 2 layers of cryptography:

- **Unique network session key shared between end-node and network server**.
- **Unique application session key shared end-to-end at the application level**.

So data are encrypted twice, the first time from the end device and the second time from the LoRaWAN protocol. When the network server has received the packets, it can use the network key decrypt the message and then it passes that message to the application server that can use the application key to obtain the original message.

CELL CAPACITY ??

2.5.6 IEEE 802.15.4

Is a standard used by **Zigbee protocol** used for low-power and low-data rate **Wireless Personal Area Networks (WPAN)**. This standard uses only the first two layers:

- Physical Layer
- Data Link Layer

In addition, it provides two more layers that are on top of the previous two:

- **Logical Link Control (LLC)**
- **Service-specific Convergence Sublayer (SSCS)**

These two layers help in the communication of the lower layers with the upper layers. This standard operate using the **ISM band**. To encode the bit it use the **Direct Sequence Spred Spectrum (DSSS) modulation**. DSSS spreads a signal over a wide bandwidth using the **spreading code**, it is a **unique sequence of digital chips that is added to the original signal to spread it over an higher frequency and wider spectrum**. The receiver and sender use the same spreading code in order to demotulate and modulate the signal. They agree on the same spreading code with a synchronization step. By using **DSSS this standard enables a wider bandwidth operation with enhanced security by modulating pseudo-random noise signal**. Furthermore, usign this standard we are more robust against noise and interferences. IEEE 802.15.4 uses **CSMA/CA to avoid collisions and also the temporal multiplexing to allow multiple access from different users**. The IEEE 802.15.4 is used only for infrequent communcations and short packet transmissions with a low duty cycle. This standard work on range distances around 10 up to 75 meters. This standard supports two types of devices:

- **Reduced Function Device (RFD)**, can talk only with FFD and have a low power consumption since has very low requirements in terms of CPU and RAM.
- **Full Function Device (FFD)**, can talk with all types of devices and support full stack protocols. They can be:
  - **Coordinator**.
  - **Router**, establishes a multi-hop connection in the network.
  - **End Device**.

The standard supports two different network types:

- **Beacon-enabled Network (slotted)**, there is a coordinator FFD which sends periodically beacon messages used to poll the other devices. These messages are used for synchronization and association of other nodes with the coordinator. The dataframes are sent usign slotted CSMA/CA with a superframe structure managed by the PAN controller.

- **Non-Beacon-enabled Network (non-slotted)**, allows continuous and asynchronous communications so the devices must be always on (no idle or standby mode allowed). The dataframes are sent using unslotted CSMA/CA.

We have 5 types of frames:

- **Beacon**
- **Data**
- **ACK**
- **MAC and Command**, used for association request, beacon request, coordinator realignment, etc...

2.5.7 Zigbee

Is a protocol which implements the IEEE 802.15.4 standard and adds **network constructions** to it. Zigbee allows to build a low cost infrastructure with low complexity which implies an extremly long battery life. The main drawback is the datarate since we are using IEEE 802.15.4. The common usage of Zigbee is the sensing and control networks. **Zigbee Alliance** is a community of vendors which has defined a protocol using the WPAN in order to create an high-interoperability between the devices. Zigbee can operate in the range of distnaces as the WiFi protocol but with a low datarate (like LoRa). **The frequencies used by Zigbee are free-unlicensed ranges and the physcal and mac layers are designed to handle multiple low data-rate operating devices**. The topologies that can be achieved with this protocol are:

- **Star**, we have a coordinator that establishes the topologies and receives all the messages between the other nodes, that cannot communicate directly with eachother.
- **Mesh**, a topology with a multi-hop communication with more router.
- **Cluster-Tree**

In a Zigbee network we must have at least one coordinator (FFD) that forms the network, it assignes the address to the end devices, manages the security, sends the beacon messages, manages the routing of the received packets (Zigbee router) and can act also as PAN controller. The end-devices typically are RFD devices which perform simple operations and communticate with the controller. The layers of the stack of Zigbee are:

- **Physcal Layer**, consist of 3 bands, each of them is composed by differnt number of channels (27, 16, 10 respectively) and operates at different frequencies and so have different data rates.

- **MAC Layer**, handles the communication synchronization using beacon message and perform CSMA/CA.
- **Network Layer**, handles operations like routing of the packets, connections and disconnections from the devices and setting up the network.
- **Application Support Sub-Layer**, is application specific and its main task is the data management services in particular:
  - Interfacing services
  - Control services
  - Bridge between network and other layers.
- **Application Framework**, provides two types of data services:
  - Provision of key-value pair used to obtain the application object attributes.
  - Generic message services.

Zigbee provides two operational mode to send data:

- **Non-beacon mode**,the coordinators and routers monitor the active state of the received data continuously. In this mode, there is no sleep mode for the routers and coordinators so is more power consuming.
- **Beacon mode**, allows the coordinators and routers to launch a very low-power sleep state, during the absence of data communication from end devices.

There are three different types of data transfer:

- **Data transfer from a device to the PAN coordinator**, the end device sends to the coordinator a message, it can optionally reply with an ACK message.
- **Data transfer from the PAN coordinator**, the end device sends a data request to the coordinator, it replies with the data and the end device sends back an ACK message.
- **Peer-to-peer data transfer**, devices are free to communicate with every other device that is in the communication range.

Each of these three methods have two variants depending if the coordinator uses or not the beacon messages.

2.5.8 RFID

Stands for Radio Frequency Identification, we have two types of devices:

- **Tag**, devices with digitally encoded data and that can be polled by the readers. They can be either **active** or **passive** if they have or do not have a own power source.
- **Reader**, more sofisticated devices that can read the tags' encoded values by sending radio queries through an antenna.

2.5.9 NFC

Near Field Communication (NFC) was developed by Philips and Sony as short-range wireless connectivity wich enables peer-to-peer (P2P) data exchange network. The communication is achieved by **magnetic induction** when the devices are close to eachother. It supports very low data rate at small frequencies. We have two types of NFC devices:

- **Passive**, do not need a power source for communicating with the NFC reader. They can simply store the information without processing it.
- **Active**, can act both as reader and as passive device.

A small electric current is emitted by the **NFC reader** which creates a magnetic field that acts as a bridge in the physical space between the devices. This technology is **fast since does not require any manual pairing** between the devices. NFC supports three information exchange modes:

- **Peer-to-peer**, the transmitting device goes active while the receiving device becomes passive. During the reverse transfer, both devices change roles.
- **Read/write**, allows only one-way data transmission. An active NFC device connects to a passive device to read-write information from-to it
- **Card emulation**,enables an NFC device (generally, smartphones) to act as a contactless credit card and make payments using just a simple tap on an NFC reader.

3 Constraint Networks

IoT devices have several constraint, they are **battery powered, requires to work without mantainance even for years, they are typically deployed in hostile and disaster areas so the existing infrastructures can be too heavy for that purpose**. We call **Constraint**

network all the networks which are focused on that constraint and have to face **low bitrate, low throughput, high packet loss and low performance with large packets due to the packet fragmentation**. The nodes that compose these networks are called **constraint nodes** (typical IoT devices). In addition to the described constraint, we also have to consider the **duty cycles** implemented to save energy. There are several types of constraint devices:

- **Class 0**, devices with very huge constraint in term of capabilities. They cannot directly communicate with the internet, they relies on gateways or proxies.
- **Class 1**, devices with constraint in processing power and memory space. They can communicate with internet but cannot employ a full protocol stack such as HTTP.
- **Class 2**, more sofisticated devices with high power budget and that support a full protocol stack.

3.1 Low-Power and Lossy Network (LLN)

These networks are composed by nodes with limited power and limeted capabilities. The IoT stack is composed by:

- **Application Layer**
- **Network Layer**
- **Data Link Layer**, the technologies used are bluetooth, BLE, Zigbee, WiFi. These technologies differ for costs, data rate, network topologies and communication technique.

With IoT devices we uses IPv6 since we have bilions of devices. **IPv6 provides a more larger addressing spaces and more routing functionalities** not provided by IPv4. The packet format is different from the IPv4 packets, in **IPv6 the header has been simplified and enlarged**. The IP assignment can be done in three ways:

- **Manual**
- **Stateful configuration using DHCPv6 protocol**.
- **Stateless configuration**, without the need of DHCP since the devices can connect to the network and generate their own addresses. IP routers periodically send router advertisment messages on the network, these messages includes usefull information such as the network prefix. By combining the network prefix and the interface address (e.g. obtained usign MAC address of the device), the device can generate its own IPv6. Then we have a duplicate detection process used to check if the IP was already assigned.

The main difficulty is due the transition from IPv4 to IPv6 since they cannot communicate with eachother. We can use **Dual-stack approach** (some routers implements both IPv4 and IPv6) or **Generic Routing Encapsulation Tunneling approach** (we can use tunneling techniques to communicate between two IPv6 networks passing through an IPv4 network). However, we must take account also that IPv6 implements security functionalities that might be to complex for IoT devices and the header size cannot naturally fit inside an IEEE 802.15.4 frame.

3.2 6LoWPAN

This is a set of standards which enables the use of IPv6 over LLN. It provides many functionalities:

- **Device addressing**, the addresses are automatically formed by the prefix of the LoWPAN edge router and the MAC address of the wireless card.
- **Routing**, it supports two different routing modalities:
  - **Mesh-under routing**, uses the MAC layer addresses to forward data packets and is usefull for smal networks.
  - **Route over routing**, uses the IPv6 addresses to forward data packets and is suitable for large networks.
- **Header extension and compression**
- **Fragmentation**, provides fragmentation in order to fit the IPv6 packets into IEEE 802.15.4 packets. When using the mesh under routing the packets are reassembled at the destination, in the route over routing the packets are reassmbled and fragmentated at each hop.
- etc...

It provides three network topologies:

- **Simple LoWPAN**, one edge router for network.
- **Extended LoWPAN**, more edge router for network.
- **Ad-Hoc LoWPAN**, no edge router.

6LoWPAN has three types of devices:

- **Hosts**, end devices .
- **Routers**, forward data inside a LoWPAN.

- **Edge Routers**, connect a LoWPAN to an external IPv6 network.

This standard uses IEEE 802.15.4 in both physcal and data link layers, on top of them it implements other ad-hoc layers.

3.3 Routing Protolc for LLN (RPL)

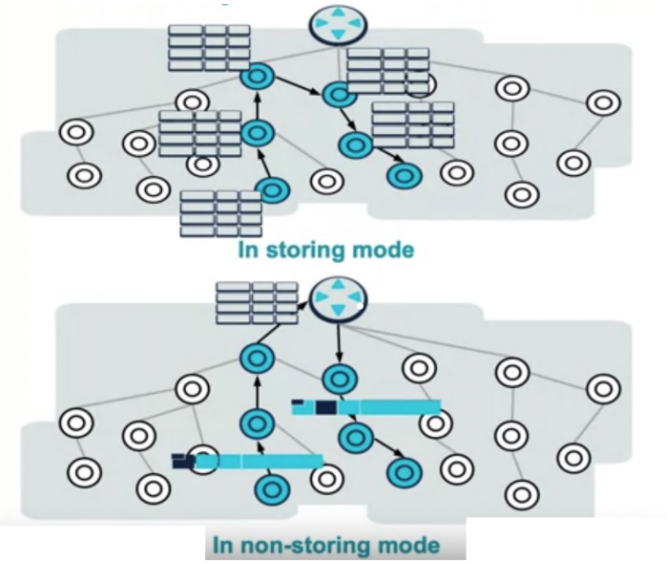
Designed for mesh under routing in LLN, so it is an IPv6-based routing protocol that works with low power transmission, large networks, lossy transmission and limited device capabilities. RPL follows **distance vector-based routing mechanism** and aims to achieve a **destination-oriented directed acyclic grap (DODAG)**, so a logical directed routing topology built over the physcal link and the **best path is selected using metrics, objective function and constraints**. The metrics typically used are the **expected transmission values (ETX), path latencies and others** (ETX indicates how many packets we have to send to deliver correctly a number of packets). All these constraint and other are dictated by the objective function and are minimize/maximized based on the metric. Some links are negleted by the graph since only the best are chosen. A node in RPL can act with multiple roles and can be part of multiple RPL graphs. In this topology we have a hierachical organization of the nodes with the RPL root (router) on top. In RPL we have:

- **Data Plane**, the network is organized in order to allow a multihop communication to reach the root (gateway). The communication can be:
  - **Point to Point**
  - **Point to Multipoint**, from one point the packet is forwarded to more nodes.
  - **Multipoint to Point**, more nodes sends packets to a single node, typically the root.
- **Control Plane**, RPL uses DODAG to find the most suitable path. Starting from the root, the nodes can communicate only with the neighbors, each node will broadcasts a message to advertice the other nodes about its presence. The parent of the node is selected by the metrics mentioned before. All the links are directed towards the root to avoid cycles in the graph.

We can have both joint and disjoint multiple DODAG graphs insde a LLN. The paths in the DODAG are categorized as **downward or upward path** (down/up respect to the root). The communication mode can be

- **Storing Mode**, each node stores the reachable nodes in its sub-DODAG.
- **Non-Storing Mode**, the root is the only node that knows the path to reach each node.

When using a point-to-point communication in non-storing mode, the sender sends the packet to the root and then the packet is forwarded to the final destination. When using a storing mode, the packet is sent to the common ancestor between it and the final destination, then the packet is sent to the destination. So in both cases we are probably not taking the shortest path, and this is called **RPL path stetching**.



To create a DODAG we can have to build the upward routes and the downward routs, we build them in a different way:

- For upward routes, the root advertices its presence using **DIO messages which contains many information such as the rank of the node and the communication mode**. The ranks increase by going down in the DODAG. After this, the other nodes also broadcasts to the neighbors their DIO messages and evaluate each link based in the metrics by selecting the parent. When a node has decided its parent, it will send to the neighbors the updated rank.

- For downward routes, are built as consequence of the upward routes creation. In fact, once that the root has sent the DIO message, it does not know about what is happening. So when the upward routes are established, the leaves propagates the information that reach the root.

4 control packets are used during the DODAG creation:

- **DIO (DODAG Information Object)**, used to establish the upward routes.

- **DAO (DODAG Advertisment Object)**, used to establish the downward routes.

- **DIS (DODAG Information Solicitation)**, used to solicitate the transmission of DIO messages.
- **DAO-ACK**

The rank are created based on the objective function based on a distance metric that must be different from the number of hops and they increase going from the root to the leaves. The metrics can be:

- **Additive metric**, such as the distance or the latency. The metric is summed when at each hop.
- **Concave metric**, such as the bandwidth. At each step we take the minimum value that act as bottleneck.
- **Multiplicative metric**, such as the loss percentage. The metric is multiplied at each step.

In a DODAG is used one of these two objective functions:

- **OF 0**, use hop count multiplied by constants.
- **OF 1**, select routes which minimize an additive metric by default the ETX so the probability that a packet is received both following upward routes and downward routes.

3.4 IoT Messaging Protocols

The session/application layer protocols provide:

- **Message abstractions**
- **Primitives for data communication/message exchange**
- **Specific networking paradigms** (e.g. publish-subscribe and request-response)
- **Additional reliability or security mechanisms**

3.4.1 CoAP

Constraint Application Protocol is a messaging protocol for constraint networks and constraint nodes. CoAP is **based on a Request/Response model (client/server)** between communication endpoints. It is a lighter and faster variant of HTTP, it **reduces the amount of exchange bytes for communication between two devices by removing some overhead**

introduced by the classic TCP/IP stack. The conversion between HTTP and CoAP is possible using REST-CoAP proxies. CoAP is designed to be used between constraint devices in the same network, between traditional devices or constraint devices in different networks. The **interaction model between CoAP is similar to the client/server model**, but unlike the classic client/server model, **CoAP handles exchanges in an asynchronous way using UDP**. The CoAP request is equivalent to that of HTTP and is sent from a client to request an action on a resource, **identified by a URI**, on a server. In CoAP we have both discovery and proxy services. CoAP implements lightweight reliability mechanisms:

- **Duplicate detection**,for both Confirmable (CON) and Non- Confirmable (NON) messages.
- **Simple stop-and-wait retransmission reliability with exponential backoff**, for Confirmable messages. The sender retransmits the Confirmable message at exponentially increasing intervals, until it receives an ACK (or RST message) or runs out of attempts.

The client can send requests to the server using its URI or sending a multicast CoAP request.

3.4.2 MQTT

Message Queue Telemetry Transport protocol is a **simple lightweight messaging protocol designed for constraint networks which uses a publish-subscribe model**. MQTT works over high latency and limited bandwidth of unreliable networks without significant needs in term of device resources. It is **built on top of the TCP protocol and uses UDP**. It implements a publish-subscribe messaging mechanism:

- **Publishers**, produce data and send them to a broker.
- **Subscribers**, subscribe to a topic of interest, and receive notifications when a new message for the topic is available.
- **Brokers**, filter data based on topic and distribute them to subscribers.

These roles are purely logical, a publisher can also act as subscriber on a different topic. **All these communications are asynchronous**.

3.4.3 HTTP

IoT can also use HTTP, in this case we talk about Web of Things. **HTTP is very standardized to is available on every device** such as computer, smartphones, etc, so we can access the **IoT things through a simple client like a browser**. A **device to implement the HTTP must be more sofisticated since they must be able to reply to HTTP request implementing**

an HTTP server. Things can transmit their sensed information through HTTP responses. The **main advantage** of using HTTP rather than other protocols, is that we can **abstract the complexity and variety** of lower-layer protocol. Devices are very heterogeneous but in this way can be accessed in the same way. The main drawback is the required things complexity since they must support TCP/IP stack.

In a typical client/server model, the server cannot send spontaneously the data to the client, it can only reply to the client's requests. In WoI we can also send data without the server being polled by the client. The REST principles:

- **Stateless**, the state is not stored by the server. They can be stored in the client or in the requests.
- **Uniform Interfaces**, each resource must be addressed using a unique address URI. Each device must have a root URL refferring to its network.
- **Cacheable**, data are cached by clients and intermediaries.
- **Layered System**, intermidiate components can hide what is behind them (proxy).

We can use any standardized file format such as **XML or JSON** to encapsulate the information that will be sent during the communication. In HTTP we can use the classic HTTP operations during the communication (POST, GET, PUT and DELETE). A deivce **MUST** support HTTP methods, JSON as default representation, have a root resource accessible via HTTP, be a server. A device **SHOULD** run HTTPS, provide human-readable documentation, support the Web Things model. **WebHooks allows the thing to act as client and send POST to other part of the communication without using polling mechanisms**. This requires that the real client uses the **Web Sockets** enabling full-duplex communication (biderictional communication).

4 Flying IoT

A **FANET** is a flying ad-hoc network, composed by a base station that collects the information coming from the flying devices. A **UAV** device is an unmanned vehicle that can be decide itslef its task or can be provided by the central authority. Can be used to monitor areas, delivering objects and patrolling target areas (similar as monitoring but with more specific tasks). The drones have processing capabilities in order to manage the movement and the monitoring tasks, obviously they are also able to communicate. Are deployed in disaster areas, mountaints, hills, etc... **FANET are fast to be deployed and can adapt to a very large amount of scenarios**. There is no standard on the communcation technique used, so we must choose it based on the application, in particular, we have to find the **right balance between availability for the application and latency for the communications**. In this networks we have to face several probelm:

- **Target coverage**, we have a set of target that must be monitored (typically a larger number than the number of drones), so we have to project the trajectories of the drones in order to maximize early coverage of the targets. The early coverage can be evaluated usign different metrics:
  - Sum of the covered targets in the round n.
  - Total covered target in n rounds.
  - Weighted progressive coverage, so the sum of covered targes but each target has an importance value.
- **Choose the right communication model**, we can use solutions like **data ferries** in which some devices monitor the area without caring about the communcation, other devices called **ferries** will ensure communication by delivering the information. The more will be the number of ferries the low will be the latency but also the lower will be the coverage since we are using a small number of the total drones to cover the areas. Another solution called **Connected Deployment** is to limit the movement of drones, they can move to a point only if they are connected (even using a multi-path communcation) with a base station. The idea is to move the drones in formations which maximize the number of target covedered ensuring that they can communicate with the base station, some of them could act as relay node without sensing nothing. We have also to minimze the time to move from a formation to the next and the time taken to compute the next formations. We must consider also the distances in order to minimize the energy consumption.