Marie Skłodowska-Curie Actions (MSCA)

Research and Innovation Staff Exchange (RISE)

H2020-MSCA-RISE-2017

Intelligence-Driven Urban Internet-of-Things Ecosystems for Circular, SAfe and IncLusive Smart CITIES

D3.1: IoT Cloud infrastructure and location sensing in urban environments

Abstract: This deliverable reviews the existing landscape of wireless IoT infrastructure options suitable for smart urban environments, starting from wireless sensors at the edge up to the networking solutions necessary for efficiently handling the large amount of data produced from such environments. Considerations for efficient power management and spectrum allocation are also included. Location sensing techniques are presented which do not rely on Global Navigation Satellite Systems (GNSS), and are thus equally suitable for both outdoor and indoor urban environments.

Contractual Date of Delivery	31/12/2019
Actual Date of Delivery	13/01/2020
Deliverable Security Class	Public
Editor	Christos Iraklis Tsatsoulis (NPS)
Contributors	NP, FORTH, CBN, BLS, BU

The *IDEAL-CITIES* consortium consists of:

FOUNDATION FOR RESEARCH AND TECHNOLOGY -HELLAS	FORTH	GR
ECOLE NATIONALE DES PONTS ET CHAUSSEES	ENPC	FR
BOURNEMOUTH UNIVERSITY	BU	UK
BLUESOFT SPOLKA Z OGRANICZONA ODPOWIEDZIALNOSCIA	BLS	PL
CABLENET COMMUNICATION SYSTEMS LTD	CBN	CY
NODALPOINT SYSTEMS	NPS	GR

This project is supported by the European Commission under the Horizon 2020 Program (2014-2020) with Grant agreement no: 778229

Document Revisions & Quality Assurance

Internal Reviewers

1. Andreas Miaoudakis (FORTH)
2. Kyriaki Konstantinou (CBN)
3. Jakub Rola (BLS)
4. Christos Iraklis Tsatsoulis (NPS)
5. Theodoros Kostoulas (BU)

Document History

Version	Date	By	Overview
0.1	9/10/2019	Christos Iraklis Tsatsoulis (NPS)	Table of Contents
0.2	14/10/2019	Christos Iraklis Tsatsoulis (NPS)	Table of Contents and contributors reassignment
0.3	1/11/2019	Christos Iraklis Tsatsoulis (NPS)	Section 2.1 (First Draft)
0.4	1/11/2019	Andreas Miaoudakis (FORTH)	Section 2.2 (First Draft)
0.5	6/11/2019	Christos Iraklis Tsatsoulis (NPS)	Section 2.3 (First Draft)
0.6	8/11/2019	Michał Kulenty (BLS)	Section 3.1. (First Draft)
0.7	8/11/2019	Andreas Miaoudakis (FORTH)	Section 2.6 (First Draft)
0.8	11/11/2019	Christos Iraklis Tsatsoulis (NPS)	Section 3.1 (First Draft)
0.9	12/11/19	Othonas Soultatos (FORTH)	Section 3.3 (First Draft)
1.0	12/11/19	Marios Angelopoulos (BU)	Section 2.5 (First Draft)
1.1	28/11/19	Kyriaki Konstantinou (CBN)	Section 2.4 & 3.2 (First Draft)
1.2	03/12/2019	Kyriaki Konstantinou (CBN)	Section 2.4 & 3.2 amendment and update (First Draft)
1.21	9/12/2019	Andreas Miaoudakis (FORTH)	Review of Section 3.1.2
1.22	9/12/2019	Kyriaki Konstantinou (CBN)	Review of Section 2.3
1.23	9/12/2019	Jakub Rola (BLS)	Review of Section 3.1.1
1.3	12/12/2019	Christos Iraklis Tsatsoulis (NPS)	Section 4 (first draft) Consolidated review (v1.23) Review of non-NPS sections
1.31	16/12/2019	Theodoros Kostoulas (BU)	Section 1 (first draft) Review of Section 2.1
1.32	17/12/2019	Christos Iraklis Tsatsoulis (NPS)	Section 5 (first draft) Section 4 (second draft) Abstract Review of Section 1

1.33	18/12/2019	Jakub Rola (BLS)	Section 3.1.3 correction
1.34	18/12/2019	Theodoros Kostoulas (BU)	Review of Abstract, Sections 4 and 5
1.35	20/12/2019	Christos Iraklis Tsatsoulis (NPS)	Consolidated review & corrections (v 1.33 & 1.34)
1.36	10/12/2019	Aristos Andreou (CBN)	Edits on Section 3.2
1.35	13/01/2020	Andreas Miaoudakis (FORTH)	Fixing references and release

Table of Contents

List of Tables

List of Figures

1 Introduction

Smart urban environments bring together a range of sensing, network, and cloud technologies in order to render the associated services to the end user.

A major backbone for offering these services is the wireless sensor network, which uses the corresponding endpoints for capturing and rendering information. In this sense, these endpoints are sensors (input) devices for capturing the different context and semantic information, and output devices for interacting with the natural environment via motion or other appropriate actions. One major constraint that should be considered when realising a smart environment is the power requirements for the employed devices. These are strongly correlated with the frequency of the data acquired by or transmitted to the sensors and actuators respectively, and define the possible battery specifications for supporting the physical infrastructure. Efficient spectrum utilization is also a concern.

The cloud technologies are employing the necessary networking equipment for supporting the sensor network and hosting a range of functionalities related to service provisioning and security, in order to meet the service level agreements. Adequate quality of service (QoS) techniques are employed in order to ensure the delivery of services related to safety or medical emergencies over less critical ones. Moreover, the required edge-to-cloud protocols are in place in order to allow the, generally constrained, IoT devices to send data over a network that imposes its own constraints.

The aforementioned infrastructure can provide a holistic support for location sensing. This should take place and be applicable in both indoor and outdoor environments, with the challenges imposed in indoor localization mainly existent due to the absence of quality GNSS signal in such environments. In this sense, visual localization can be used to visually search through a location-tagged image database and return the closest image to the query one, along with its exact location, independently of whether this corresponds to and indoor or an outdoor location.

2 Wireless Sensor Networks

2.1 Network Endpoints

2.1.1 Sensor nodes

Sensor nodes form the interface between our analogue reality and the digital realm, measuring natural phenomena and converting them to processable units of information. There are many different categories of sensors for IoT networks, according to the sensor type and/or application domain. According to Rosza et al. [1] they can be classified in five groups:

- **Motion:** sensors measuring the movement of a body (solid, liquid or gaseous; animate or inanimate)
- **Position:** sensors measuring the position of a body (solid, liquid or gaseous; animate or inanimate)
- **Environment:** sensors measuring inputs originating from the environment
- **Mass Measurement:** sensors measuring the interaction between a body with another body, or a body with another environmental force
- **Biosensor:** sensors used for retrieving measures from organisms

Within these groups, sensors can be further categorized according to their function. A non-exhaustive summary of the various functional categories per group, as per Rosza et al. [1] is shown in Table 1:

Table 1: Sensor Categories per Group

Motion	Position	Environment	Mass Measurement	Biosensor
Movement	Orientation	Temperature	Volume	Blood
Velocity	Inclination	Humidity	Pressure	Organ
Inertia	Proximity	Luminance	Density	Mental
Vibration	Presence	Acoustic	Deformation	Tissue
Acceleration	Location	Radiation	Viscosity	
Rotation		Gas	Flow	
		Weather	Load	
		Chemical	Moisture	
		Electrical	Shock	
		Color	Contact	
		EM Field	Strain	
			Corrosion	
			Electrical Conductivity	
			Oxygen	

Further to different functional categories, different types of sensors exist within each category. Often, they rely on different physical concepts to provide measurements, and exhibit differences with regards to usable features, size, durability, and accuracy among others. This affects the cost of the sensor, which has to be weighed against the benefits arising from the usage scenario. Consequently, the sensor types to be used within the context of the

project need to be carefully considered on a per-case basis. Two important sensor categories with respect to potential project applications are Luminance and Movement. They are described in the following subsections.

2.1.1.1 *Light Detection*

Detecting the existence (or absence) of light and its intensity is an important building block towards adding intelligence to an urban environment, e.g. for enhancing citizen safety or for providing smart lighting service in some areas. The common ways to translate light to electrical signals are photoresistors (Fig. 1), which change their resistance depending on light intensity, and photodiodes, which convert light to electrical current. When choosing a sensor, several key factors need to be considered such as light sensitivity, active power supply & power consumption, sensitivity to temperature changes, sensing distance, or the elapsed time until a change of light is converted to an electric signal.

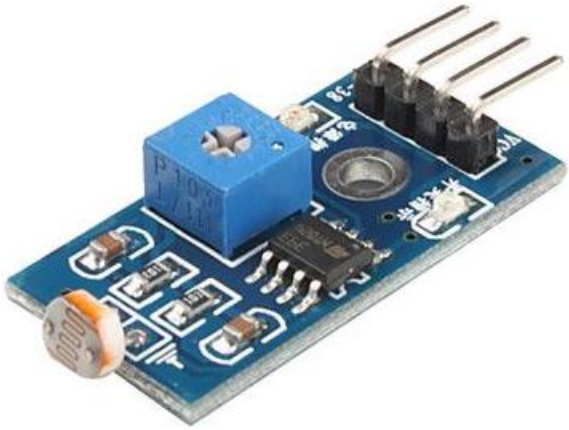

Figure 1: Photoresistor Sensor Module

2.1.1.2 *Movement Detection*

The common sensor types for movement detection are passive infrared (PIR) and microwave (Fig. 2). PIR sensors enclose material sensitive to heat and infrared waves, placed below a lens (or usually a set of lenses) to widen the field of view of the sensor. When the material is subjected to IR radiation, it creates electric current, which in turn is amplified and sampled. Factors to consider are sensing range, sensing angle, and wavelength sensitivity (e.g. 8-14 μm for human body detection).

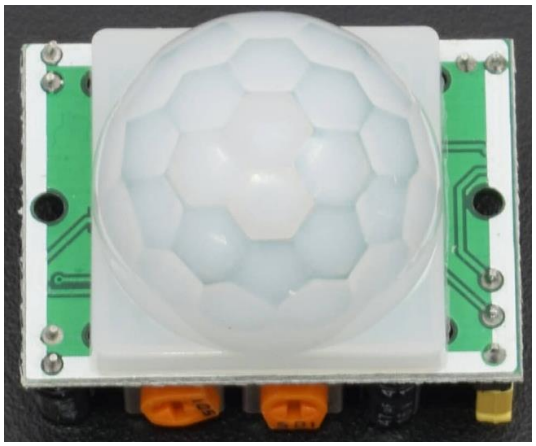

Figure 2: PIR Sensor HC-501SR

The affordability, deployment simplicity, and low power requirements of PIR sensors have made them a popular choice for a wide range of applications, however there are limitations to their usage, which need to be accounted for in the project context. Perhaps the most critical limitation is that PIR sensors require a line of sight to properly detect a moving object, and their accuracy is affected by environmental conditions such as higher ambient temperatures, excess light, dust, vapor, smoke, or similar phenomena which obstruct visibility.

In contrast to PIR sensors, microwave sensors (Fig. 3) operate by sending out microwaves, and registering their reflection once the waves bounce back from objects. The sensor will utilize the Doppler Effect to detect if the object is moving.

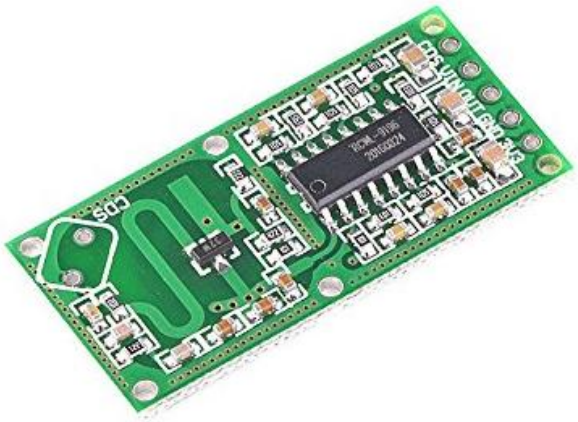

Figure 3: Microwave sensor RCWL-0516

The main benefit of microwave sensors is that they can be used in any environment, i.e. they are not constrained by heat or low visibility conditions. Furthermore, microwaves can penetrate walls, and they are not solely confined to detecting motion from heat emitting sources, like human bodies. In general, microwaves offer higher coverage range and sensitivity than PIR sensors, however they are costlier to operate energy-wise.

2.1.2 Vision Systems

Vision detection forms the backbone of one of the project's two use cases, making the selection of the related IoT devices of crucial importance to the project's success. Contrary to the aforementioned sensors, a vision capturing device is a complex system consisting of optics, electronics, and image processing capabilities. Actual sensing occurs either via charged-coupled-devices (CCD) or complementary metal-oxide (CMOS) devices. The former traditionally provided higher quality images at the expense of considerably more power, while the latter provided images of lower quality and more noise, but is much more power efficient.

Advances in CMOS manufacturing technology, especially riding on the wave of mass smartphone camera development, have significantly boosted CMOS image quality, thereby surpassing CCD-based devices in almost every performance parameter [2]. Especially in the machine vision domain, which requires high speed of image processing and low noise in image artefacts to ensure accurate results, the massively parallel & low bandwidth CMOS A/D conversion approach is superior to CCD. Consequently, vision devices with CMOS-based sensors are considered for the project use cases. This is further strengthened by the recent surge of low-cost CMOS-based vision systems that integrate HD video capabilities, IR night vision, and high compression & storage with an IP-based WiFi access point for easy installation and operation.

Figure 4: GoComma CA-R20A Wireless IP Camera

Factors to consider when choosing the vision devices will be image resolution, frames-per-second and power consumption. Due to the large amount of data produced by these devices during operation, processing of this data will occur as close to the device (edge) as possible.

2.1.3 Output Devices

An IoT output device can be any form of manifesting information, such as a LED light, a sound indicator, or a monitor. An important family of output devices are actuators, which have the ability to create motion and physically interact with the environment. Actuators can be categorized by the way they create motion; notable categories are:

- Pneumatic actuators:
- Electrical actuators:
- Hydraulical actuators:
- Thermic actuators

In addition to how motion is created, the type of motion of an actuator defines its construction type. The main types are:

- Linear actuators
- Motors
- Relays
- Solenoids

2.1.4 Power considerations

When deploying a wireless sensor network, significant attention needs to be directed to the proper power management of the constituting nodes. IoT devices will almost always be battery operated; care must be taken to optimize the usage patterns to preserve power and prolong device operation. This is a critical point in order to avoid complex and costly maintenance logistics, given that:

- The number of deployed IoT sensors and devices can potentially be very large,
- IoT devices and sensors may likely be in remote areas
- IoT devices and sensors may not always be easily accessible

The different role and importance of each IoT device in the project context warrants a per-case power management approach. Aspects to consider when planning device operation are:

- Power required by sensor / device during active operation
- Power required by sensor / device during standby
- Power required by device for radio communication
- Power required by actuators located on the device (e.g. a stepper motor to move the camera lens)
- Frequency of data collection
- Frequency of data transmission
- Battery capacity

2.2 Efficient spectrum utilization approaches

Mobile services already generate the greatest economic value by some distance - 269 billion in the EU27 in 2013 [3]. The next most valuable application is civil aviation, with an economic value of 159 billion. It is expected that this gap will increase over the next 10 years as the economic value of mobile services grows to a forecasted 477 billion in 2023. This growth is fueled by two main developments:

- The central role that mobile services will play in the provision of broadband services. As end-users move to smart phones and operators roll out LTE, we will see a dramatic increase in the functionality which end users enjoy when using mobile services
- The development of the Internet of Things (IoT), where mobile networks are likely to provide an important transport mechanism for machine-to-machine communications. This development is uncertain, but could generate just under €90 billion of additional economic value in 2023

The exponential growth of wireless communication systems and services increases the need for wireless spectrum, which is expected to grow very rapidly. Nevertheless, the radio spectrum is already very crowded, as most of the available frequency bands are already permanently reserved for dedicated wireless communication networks and exclusive use. Additional services and systems appear to be very hard to find space to fit in the crowded wireless spectrum.

On the other hand, it is very common that allocated spectrum usage is under-utilized mostly due to the inflexibility and ineffectiveness of the static spectrum allocation policy worldwide. Typically, a portion of the licensed bands are allocated in a specific time and area. Several studies ([4], [5]) show that the fixed radio frequency allocation may lead into significant underutilization of the radio spectrum due to very sporadic usage across different geographical regions, as well as in different periods of time [6]. Such results dictate the reconsideration of the traditional centralized administrative spectrum management approach, opening space for more efficient wireless communication systems based on the novel dynamic/opportunistic spectrum access.

The key to solving problems of inefficient spectrum usage lies in the concept of Dynamic Spectrum Access (DSA) [7]. DARPAs approach on Dynamic Spectrum Access network, the so-called NeXt Generation (XG) program, aims to implement the policy-based intelligent radios

known as cognitive radio. Cognitive Radio (CR) is a promising solution to the problem of overcrowded/underutilized spectrum. CR is an intelligent radio technology that has the capability to search and utilize underutilized spectrum resources [8]. CR has been recognized as an effective solution to the dilemma introduced by the rapid growth of wireless communications and the scarcity of spectrum resources [9]. Its principles are further described in Section 2.6.1 below.

2.2.1 Opportunistic Spectrum Access

Driven by national and international regulations and legislation, as well as by the current technological advancements in the radio technology, the so-called Opportunistic Spectrum Access (OSA) has the potential to mitigate spectrum scarcity and meet the increasing demand for spectrum. The principle in OSA is to enable unlicensed users, often referred as Secondary Users (SU), to dynamically access the licensed spectrum that Primary Users (PU) are licensed to use when licensed bands are not used (white spaces). OSA is leveraged by CR, since an essential required functionality is the spectrum usage sensing. However, the power of CR radio by itself cannot make OSA succeed. CR networks can only dynamically access the white spaces when PUs are absent. The protection of the PUs communications is a key issue on the research of cognitive radio network spectrum in an opportunistic fashion.

Spectrum sensing and so-called listen-before-talk strategy are vital for protecting PUs. Due to the significant importance of spectrum sensing in CR technology, it has grown to become a very active research topic over the last years. The outcome of this research effort is the proposal of several spectrum sensing techniques based on the Energy Detection (ED) [10], Matched-Filter Detection (MFD) [11], Cyclostationary Feature Detection (CFD) [12], Covariance-Based Detection (CBD) [13], Wavelet-Based Detection (WBD) [14], and Compressed Sensing Detection (CSD) [15]. These techniques have different implementation requirements, and therefore can be classified into four main categories:

- Techniques that require a priori information of the PUs signals (e.g., MFD and CFD)
- Techniques that require no a priori information of the PUs signals (e.g., ED, WBD and CSD).
- Techniques that can sense the narrowband PUs signals (e.g., MF, CFD and ED)
- Techniques that can sense the broadband PUs signals (e.g., ED, WBD and CSD)

Of course, each one of these approaches has advantages and disadvantages. For example, the use of the energy detection for spectrum sensing is limited due to its poor performance in low SNR environments. Similarly, the use of the matched filter detection for spectrum sensing can be also highly limited since it requires a-priori information of the primary user's signals which may not be available at the cognitive radio side in real wireless communication scenarios. Also, because the cyclostationary feature detection for spectrum sensing requires a high computational complexity and a priori information of the primary user's signals, its practical use is also limited. Other spectrum sensing methods also suffer from various drawbacks that can greatly limit their practical applications in cognitive radio environments

2.2.2 Wireless-link property improvements

OSA approach leveraged by CR capabilities may also be exploited in wireless sensor networks (WSNs), which are traditionally assumed to employ fixed spectrum allocation, and characterized by the communication and processing resource constraints of low-end sensor

nodes. Depending on the application, a WSN with CR capable sensor nodes can improve their performance by taking advantage of features as:

- Dynamic spectrum access: in conjunction to the existing WSN deployments that have a fixed spectrum allocation over very crowded unlicensed bands, DSA can provide efficient bandwidth resources without the cost of spectrum lease for licensed bands.
- Opportunistic channel usage for traffic burst: It is common that a large number of sensor nodes generate bursty traffic that need to propagate to the cloud. This increases the probability of collisions and packet losses, which decreases the overall communication reliability with excessive power consumption. Opportunistic access to multiple alternative channels may alleviate these potential challenges.
- Adaptability for reducing power consumption: The dynamic nature of the wireless channel is responsible for packet losses and therefore retransmissions. Cognitive radio capable sensor nodes may be able to adapt to varying channel conditions, which would increase transmission efficiency, and hence help reduce power used for transmission and reception.
- Overlaid deployment of multiple concurrent WSN: Dynamic spectrum management may significantly contribute to the efficient coexistence of spatially overlapping sensor networks in terms of communication performance and resource utilization.
- Communication under different spectrum regulations: A certain band available in one specific region or country may not be available in another due to varying spectrum regulations. Sensor nodes equipped with cognitive radio capability may overcome this potential problem.

In general, a cognitive radio wireless sensor network (CRWSN) can be defined as a distributed network of wireless cognitive radio sensor nodes, which sense event signals and collaboratively communicate their readings dynamically over available spectrum bands in a multihop manner to ultimately satisfy the application-specific requirements. However, the realization of CRSNs and their potential advantages outlined above depends on addressing challenges including inherent resource constraints of sensor nodes, additional communication and processing demands imposed by cognitive radio capability, design of low-cost and power-efficient cognitive radio sensor nodes, and multi-hop opportunistic communication over licensed and unlicensed spectrum bands in densely deployed sensor networks.

2.2.3 Network protocol design

CRWSNs operate in an environment where they should use the available white spaces. To support such capability of CRWSNs, the redefinition of the protocol stack is required introducing some new communication protocols or mechanisms for efficient spectrum utilization and of course to preserve the rights of Spectrum usage by PUs. Protocol designs for CRWSNs MAC protocol design can be classified in a distributed and centralized classes. Moreover, there is a standardization effort by the IEEE802.22 working group [16].

2.2.3.1 *IEEE802.22 WRAN*

The IEEE 802.22 WRAN standard that has been developed with the aim of using geographically unused TV spectrum by using cognitive radio technologies while ensuring no interference to incumbent users. This standard specifies an infrastructure -based network, which is point to

multi-point, and the network is formed by a Base Station (BS) and Consumer Premise Equipment (CPEs). The BS is fully responsible to control the medium access.

One of the main functionality that 802.22 defines is the channel management. A Spectrum manager is integrated in BS with the task to manage the channels according to the reports from CPEs. CPEs report channel state, incumbent users' activities, coexistence information etc. 802.22 standard classify channels into available and not available. Moreover, the available channels are classified as:

- Operating channel: currently used by BS,
- Backup channels: potential operating channels in case of incumbents' appearance,
- Protected: prohibited by local regulations
- Candidate channels: candidates for backup channels,
- Occupied channels: currently used by other WRANs and
- Unclassified channels

Channels that are currently used by TV transmitters are classified as unavailable channels. BS maintains all the available channel sets and each CPE maintains only the first three channel sets. These channel sets are updated periodically.

Another very important feature in 802.22 is the Incumbent protection. Incumbent users represent authorized users of the spectrum like analog TV, digital TV and wireless microphones. In order to safeguard the incumbent users, 802.22 system needs information about the TV channels utilization. This is possible by two different approaches, incumbent database and spectrum sensing. Incumbent databases contain information of incumbent users' spectrum usage and maintained by regulatory bodies. Spectrum sensing can be done by both BS and CPEs. Normally, spectrum sensing is done within quiet periods which are scheduled by BS. However, BS may request to specific CPEs to perform spectrum sensing in normal operation

To reduce co-existence effect 802.22 introduces the Coexistence Beacon Protocol. This protocol is used for neighbour discovery, control information exchange among WRANs cells and inner-cell communications. Coexistence beacons are transmitted in a special time slot called self-coexistence window (SCW) including channel information and specific time schedules. If a CPE receives a coexistence beacon from other WRANs cells, it decodes it and reports to its BS. Once BS receives the report, it changes its operating mode from normal to coexistence mode.

2.2.4 Energy balance

In WSN, energy conservation is of paramount importance directly affecting network performance and lifetime. Energy consumption should be considered in the design both of protocols and hardware. Energy efficiency or consumption is directly affected by packets collision and idle listening [17]. Energy efficient design is required at every layer of the communication protocol stack. On a CR capable physical layer additional functionalities are required such as spectrum sensing and reconfiguration of the transmission parameters (such as carrier frequency, transmission power and modulation scheme. In the mac layer, flow control is implemented having the challenging task to provide fair and efficient medium access to every single node in densely deployed and resource-limited WSNs. Energy dissipation can

be reduced by introducing sleep period in the MAC scheme and adopting efficient collision avoidance mechanisms. Moreover, energy efficiency can be improved further by speeding up the convergence of network redundant data or choosing energy-efficient routing to forward data with multi-hop method.

2.3 Non-IP WPANs

A widely used classification of the wireless network technologies for IoT devices relates to the way the device is addressed. Sensor devices which do not have an IP address are part of a non-IP Wireless Personal Area Network (WPAN). Non-IP WPANs have important advantages compared to their IP-WPAN counterparts with regards to power consumption and cost. This section presents candidate technologies to be considered for the project and examines their frequency and power consumption characteristics.

2.3.1 Bluetooth

The Bluetooth wireless technology was originally conceived in 1994 by Ericsson and first released in 1998 (Bluetooth 1.0). Since then, there have been numerous revisions to the protocol, with version 5 released in 2016. By then, the Bluetooth Special Interest Group had grown to over 30000 members. Bluetooth entails two different operating technologies: BR/EDR (Basic Rate / Enhanced Data Rate), also known as Bluetooth Classic, and BLE (Bluetooth Low Energy). The chosen technology impacts interference and power consumption.

2.3.1.1 *Networking*

Bluetooth operates in the ISM (Industrial Scientific Medical) band, with frequencies between 2.4 and 2.4835 GHz. In order to mitigate interference, Bluetooth utilizes the Frequency Hopping Spread Spectrum (FHSS) technique, where communication occurs on a predetermined sequence of distinct frequencies known to the transmitted and receiver. These distinct frequencies are referred to as channels, and their number varies depending on the operating technology; Bluetooth classic uses 79 channels, while BLE requires 40. The hopping rate is set at 1600 hops per second. In addition to FHSS, Bluetooth employs another technique to combat spectral interference known as Adaptive Frequency Hopping (AFH). In AFH, nodes can detect which channels exhibit high interference and switch to another frequency hopping pattern consisting of less noisy channels. A typical case where AFH comes into play is when one device transmitting at a high bitrate, e.g. a recording video camera, takes up multiple channels for streaming data during a prolonged period of time. Here, other devices sharing the same frequency band will avoid these channels by using AFH to switch to unused channels.

The channel scheme differs between Bluetooth Classic and BLE:

- BR/EDR:
    - Has a channel center frequency of each 2.402 GHz + k * 1 MHz, where k is between 0 and 78.
    - Uses Time Division Duplexing (TDD) for full duplex communication
    - Reaches 1Mbps raw data rate in BR mode, and up to 3Mbps in EDR mode.
- BLE:
    - Has wider channels than BR/EDR, with a channel center frequency of each 2.402 GHz + k * 2 MHz, where k is between 0 and 39.

- Three of channels out of 40 are reserved for advertising
- Uses Time Division Multiple Access (TDMA) for bidirectional communication
- Reaches 1Mbps raw data rate (2Mbps for Bluetooth 5)

Further to channel usage, additional distinctions concerning the networking modes exist between BR/EDR and BLE. Although both technologies use a master-slave communication model, where the node initiating the connection becomes the master and the node advertising an available connection becomes the slave, there are notable differences with regards to network topology. In the BR/EDR case, a network ("piconet") is formed between a master and up to seven slaves due to the assigned device networking address having a length of 3 bits. If a slave node of one piconet is simultaneously a master node of another piconet, the resulting network is called a scatternet. Contrary to ER/BDR, in BLE the networking address has a length of 24 bits, thus allowing a master to have millions of slaves. However, a master can only pair with one slave per piconet. In practice, this translates to a master having as many piconets as there are associated slaves, and BLE piconets being significantly smaller. In addition to point-to-point communications via piconets, Bluetooth 5 has introduced mesh topologies for BLE for improved reliability, which need to be taken into consideration for usage scenarios requiring delivery reliability.

Additional differences to consider are actual data throughput, which is considerably lower in BLE due to the protocol overhead, and transmission latency, i.e. the time required for the device start sending data. The main aspects are summarized in Table 2.

Table 2: Comparison of Bluetooth BR/EDR and BLE for application usage

Type	Throughput	Latency	Characteristics	Sample Applications
BR/EDR	Up to 2.1Mbps	100ms	connection oriented, high throughput	Streaming of High-Fidelity data, e.g. Wireless Headphones, Vision systems
BLE	Up to 0.27Mbps	< 10ms	low throughput, low latency, can wake-up on demand	Body sensors e.g. Heart rate, blood pressure Proximity-aware location-based beacons e.g. indoor positioning, targeted information Real-time monitoring sensors, e.g. for industrial manufacturing

2.3.1.2 *Power considerations*

In order to optimize power usage, the transmitting power of the devices should be adjusted according to its usage scenario requirements. The Bluetooth standard specifies four different power levels for device operation, summarized in Table 3.

Table 3: Bluetooth power levels

Type	Power	Max. Power Level	Max. Range	Sample Devices
Class 1 (BR/EDR only)	High	100mW (20dBm)	100 m	Access points
Class 1.5 (BLE only)	Medium-High	10mW (10dBm)	30 m	Beacons, wearable sensors
Class 2	Medium	2.5mW (4dBm)	10 m	Mobile devices, Smart Card Readers
Class 3	Low	1mW (0dBm)	1 m	Bluetooth adapters

In contrast to BR/EDR, BLE allows devices to conserve energy by activating the transmitter only if data is too be sent/received. Also specifically for BLE, Bluetooth provides a beaconing technology where designated beacon nodes periodically advertise information. Advertising nodes do not pair with any other nodes (as this would effectively stop the beaconing process); instead, they share information via broadcasting their Universally Unique Identifier (UUID). The UUID may contain additional information which may trigger actions on receiving devices. Within the project context, beacon devices can be used for broadcasting location-based information about a fixed point of interest to devices in range, or aiding of indoor positioning. For the latter, the beacon can also emit its signal strength, thus allowing the receiver to calculate the distance to the beacon by measuring the received strength.

Due to the continuous broadcasting of beacon signals, the frequency of advertising will affect the power consumption of the device and needs to be carefully considered with respect to usage scenarios requirements. More frequent advertising may result e.g. in increased accuracy with respect to location tracking, but will negatively affect the longevity of a battery-powered device.

2.3.2 802.15.4

802.15.4 is an IEEE standard which targets the physical (PHY) and the medium access (MAC) layers, designed for low-cost / low-complexity devices with low speed requirements.

The standard has served as base standard for various other higher layer technologies, including ZigBee, 6LoWPAN and others.

2.3.2.1 *Networking*

Similar to Bluetooth, 802.15.4 operates in the unlicensed spectrum using three different bands. In practice, the 2.4 GHz band is the most widely used due to its higher speed, which in turn conserves power due to shorted transmission / reception times. The range of 802.15.4 can reach 200 meters during open-air, line-of-sight propagation, and roughly 30 meters in an indoor environment. Table 4 lists the 802.15.4 frequencies and throughputs.

Table 4: 802.15.4 Frequencies & Throughput

Frequency Bands (MHz)	Channels	Throughput (Kbps)	Region
868.3	1	10, 100, 250 (depending on modulation)	Europe
902-928	up to 30 / 2MHz separation	40, 250 (depending on modulation)	North America & Australia
2405-2480	16 / 5MHz separation	250	Worldwide

Interference avoidance is accomplished using the Carrier Sense Multiple with Collision Avoidance (CSMA/CA) mechanism, which is also used in the 802.11 WiFi standard. In this mechanism, the transmitter listens to a channel and starts transmitting after a predetermined period of time if the channel is idle.

The 802.15.4 standard supports two types of devices and two network topologies. The topologies are:

- Star topology: The simplest form of an 802.15.4 network, requires one node to act as a coordinator, and all other nodes communicate through the coordinating node.

- Peer-to-Peer topology: In this mesh-type topology, all nodes can communicate with neighbouring nodes directly without the mediation of a coordinating node.

In both topologies, the coordinating node is responsible for setting up the network and sending out beacons when operating in beacon-based mode (see next section). Extensions to the 802.15.4 standard (e.g. ZigBee) allow for more complex network topologies (e.g. clusters). Network topologies also dictate the device type of the participating nodes. The possible device types are Full Function Devices (FFD), which can act both as a network coordinator and a simple network participant, and Reduced Function Devices (RFD) which can only participate in a Star topology network, without being able to coordinate it. Within a project context, a Star topology can be used when covering geographically constrained areas with low-cost sensors, whereas if more reliability and larger area coverage is required, a peer-to-peer topology can be considered. However, peer-to-peer to networks are more complex to maintain and can introduce security vulnerabilities. A table detailing benefits and considerations for usage of 802.15.4 topologies in project scenarios is shown in Table 5.

Table 5: 802.15.4 Topologies comparison for applications

Topology	Benefits	Considerations
Star	Cheaper devices to due to usage of RFD nodes Easier to setup and maintain	Limited coverage area due to one hop communication Coordinating node has to be technically more complex and secure in order to act as a hub

		Coordinating node can be a single-point of communication failure Lower bandwidth available
Peer-to-Peer	Increased coverage area Increased network robustness More communication bandwidth Better communication reliability	Larger attack surface can make the network more prone to security incidents Higher power consumption More expensive FFD-type devices required More difficult to set up and maintain

2.3.2.2 *Power considerations*

The 802.15.4 standard mandates a minimum transmit power of 3dBm, and receiver sensitivity of -85dBm for the 2.4 GHz band. The actual power usage depends on the how the device is operated. Here the standard defines two modes of operation:

- Beacon-based: The coordinating node periodically sends out a beacon, which is picked up by the participating nodes. A participating node responding to the beacon can signal its intent to use the channel, and the coordinator can then allocate a dedicated, contention-free time-slot for the participating node to transmit. A typical usage scenario for energy conversion is for participating nodes to be in a sleep state, and periodically wake up to listen for beacons.

- Beaconless: During this mode of operation, all nodes are listening to the channel. While this operation is simpler, it also needs to ensure that nodes access the communication channel when it is clear of traffic. It should be noted that channel interference could result also from devices that are not part of the 802.15.4 network (e.g. a Bluetooth device). A special activity specified as Clear Channel Assessment (CCA), is undertaken by each node wishing to transmit, in order to sense if the channel is used. Given that all nodes are always listening to the channel (i.e. are in receiving mode) and CCA needs to be performed prior to transmitting, the beaconless mode of operation is consumes significantly more power than its beacon-based counterpart.

Within the project context, a beacon-based communication will be preferred, especially in areas with high interference from other WPANs, and when no critical real-time sensor data is required.

2.3.3 ZigBee

As stated in the previous section, ZigBee is a communication protocol which is based on the 802.15.4 specification for its lower PHY and MAC layers. However, the protocol has been designed primarily with mesh networking in mind, therefore it provides additional functionalities in the networking layers in order to provide advanced capabilities such as dynamically forming or self-healing networks.

2.3.3.1 *Networking*

The allocated frequency bands for ZigBee stems from the 802.15.4 specification and are therefore identical. Slight differences may exist with regards to throughput in the lower

frequency bands may exist, given that ZigBee is an offshoot of the older 802.15.4 specification released in 2003, and since then the PHY and MAC layer in 802.15.4 has been revised. The main difference to 802.15.4 is how ZigBee treats the networking services on top of these layers. Specifically, ZigBee divides the nodes into three types:

- Controllers: Similar to the 802.15.4 network coordinator, this type of node is responsible for initiating the network and assigning addresses to participating nodes.

- Routers: Of interest especially for mesh networking, this type of component handles routing of messages by maintaining routing tables. It can also add also assign addresses and allow new nodes to join the network. Its existence is optional.

- End Devices: These are the ZigBee equivalent of the 802.15.4 Reduced Functional Devices (RFDs) and can only communicate with controller router nodes. This is also the only node type which can enter sleep mode.

ZigBee uses two types of addresses, long, 64bit addresses unique to the device (and the greater network), and short 16bit addresses which are dynamically assigned by a controller or router node to devices when joining a network. As a consequence, the number of endpoint devices per controller/router nodes is limited to 240, similar to the TCP/IP protocol. With regards to supported topologies, ZigBee provides three options:

- Star topology: Practically identical to the star topology specified in 802.15.4, this topology requires one central coordinator node, and all communication needs to occur via this node. This is the simplest topology and the most easy to maintain.

- Cluster Tree topology: This topology can be imagined as a hierarchical arrangement of smaller star networks, with the controller node being at the top, and with router nodes connected to the controller as child nodes, forming a small sub-network of End Devices and/or other routers. Essentially, additional levels are added to the network, thus extending its coverage. However, the top level controller node can become a critical single-point-of failure, given that for communications to some (or even all) sub-networks, the controller must be traversed.

- Mesh topology: In this topology router nodes are directly connected to each other, thus routing can freely occur through any router/controller node in the network. As with the peer-to-peer topology of 802.15.4 described in the previous section, its strength lies in greater area coverage and reliability due multiple routing paths, without necessarily including the controller node.

From the aforementioned topology options, it becomes evident that ZigBee becomes a strong contender if more complex and dynamic topologies are required. Routing is an enabling part of the topology process and the available routing options are:

- Broadcasting: Enables a node to communicate with multiple other nodes using a single request. Only controller and router nodes can repeat broadcast requests. Given that this is a very resource intensive form of communication, it should be used sparingly.

- Mesh routing: One-to-one communication based on routing tables containing all discovered nodes. This is the preferred routing option in ZigBee.

- Tree routing: One-to-one communication based on routing tables for sub-networks. More memory efficient than mesh routing given the smaller routing table size, but less resilient in case a link to a router node breaks.

- Source routing: Primarily used when a "data concentrator" node (usually the gateway) wishes to send messages to hundreds or thousands of nodes in the network. Here, a node with enough memory will store the routes to all nodes and send the route to the destination node to the other nodes, which will relay the message accordingly.

From a project application perspective, it is important to understand how the routing characteristics shape the project usage scenarios in terms of area coverage and spectral efficiency. This is summarized in Table 6, adapted from [18].

Table 6: ZigBee routing characteristics

Routing method	Max. Hops	Destination	Bandwidth Efficiency	Payload Efficiency
Broadcast	30	Multiple nodes, unacknowledged reception	Low	High
Mesh	30	Single node, acknowledged reception	High	High
Tree	10	Single node, acknowledged reception	High	High
Source Routing	5	Single node, acknowledged reception	High	Low

2.3.3.2 *Power considerations*

In order to be ZigBee certified, a battery-powered device must exceed a two-year battery-life. In practice, ZigBee battery-powered devices can last for the shelf life of batteries (5-7 years for AA batteries), although this is primarily dependent on the chosen topology. In general, the more complex the network, the higher the power requirements due to the additional sensing and routing work that needs to be performed by the controller and router nodes.

The most power efficient option is a simple star topology containing only RFD devices and an FFD controller typically powered by mains. The cluster tree topology comes next, given that there must exist at least as many router nodes as there are sub-networks, thus increasing the required FFD devices. Finally, the mesh network is the least power efficient, as all router nodes must actively sense the network, maintain routing tables and manage routing. The RFD devices can typically remain in a dormant state given that they do not have to handle routing, and only wake up periodically or if there is something to transmit. Therefore, a usage specific trade-off has to be made between real-time transmission of events and RFD battery life. It should also be noted that for parent-to-child communication (e.g. controlling an RFD actuator), the parent node buffers the message for the child until the child wakes up and is ready to receive the message. The length of time for which a message is buffered by the parent

depends on the layer that is used for buffering. If the message is buffered at the MAC layer, it is usually discarded after 7 seconds. Thus, criticality of message reception needs also to be taken into account when budgeting the power requirements for the project scenarios.

2.3.4 Z-Wave

Z-Wave's principal field of usage is in consumer home automation, but due to its reliability, it is also found in commercial and industrial buildings. It was originally designed for lighting control in 1999. Twenty years on, there are over 2600 Z-Wave certified products [22].

2.3.4.1 *Networking*

Contrary to Bluetooth and 802.15.4-based technologies like ZigBee, Z-Wave does not operate in the higher ISM frequencies, but uses the short range narrow-band sub GHz bands as laid out by the G.9959 recommendation by the International Telecommunication Union (ITU). The smaller carrier wave length negatively affects data throughput, however it benefits the range of the signal, which can reach up to 100 meters outdoors, and using its ability to permeate obstacles like walls, up to 30 meters indoors. Z-Wave uses single channel frequency bands depicted in the table below.

Table 7: Z-Wave frequency bands

EU Frequency Bands (MHz)	Channel spread	Throughput (Kbps)
869.85	400kHz	100
868.40	300kHz	40
868.40	300kHz	9.6

Like the 802.15.4-based technologies, Z-Wave manages channel allocation using the CSMA/CA mechanism. Collision avoidance is accomplished by sensing the channel in receive mode, and delaying their transmission for a random number of milliseconds in case they need to transmit. A Z-Wave network is of a mesh-type, where every node is connected to its neighbour. The protocol defines two different types of nodes:

- Controllers: Manages the routing table for the network. Controllers are further differentiated in primary and secondary. A primary controller maintains the network hierarchy, computes the routing table, can communicate with all nodes and can add/exclude nodes to/from the network. There can be only one primary controller in a z-wave network. Secondary controllers participate in the routing process.

- Slaves: These devices execute the commands received by the controller. They do not compute routing tables, however they can store them and forward them to neighbouring nodes.

Routing in Z-Wave occurs through the propagation of a routing table throughout the network, wherein each node only registers its immediate (i.e. single-hop) neighbour. With regards to addressing, Z-Wave uses a much simpler scheme compared to other protocols like Bluetooth or ZigBee. There are two types of addresses:

- Home ID: A 32 bit identifier of a Z-Wave network, which is pre-programmed by the factory for controller nodes, whereas for slave nodes it is assigned by the controller.

- Node ID: An 8-bit unique identifier of a device within a Z-Wave network. For controller nodes it usually has the value of 1, whereas for slave nodes it is assigned by the controller. One network can have up to 232 devices.

2.3.4.2 *Power Considerations*

Z-Wave has been designed with power conservation and reliability in mind. This is reflected in the choice of frequency, the addressing simplicity and the basic routing scheme. Nevertheless, additional power efficiencies can be gained by operating slave devices in sleep mode, where they can be programmed to wake-up and transmit after pre-defined time intervals. However, similar trade-offs as described in section 2.3.3.2 need to be taken into account. Controller nodes cannot be operated in sleep mode.

2.4 IP-WPANs

In contrast to the non-IP WPANs presented in Section 2.3, IP-WPAN devices are network endpoints that can be addressed with an IP. The efforts of the groups working to develop IP-enabled WPAN devices is to bring the internet protocol even to the smallest devices of the IoT ecosystem.

2.4.1 6LoWPAN

6LoWPAN stands for “IPv6 over Low-Power Wireless Personal Area Networks”. It is based on IEEE 802.15.4 standard and is developed by the 6LoWPAN group of the Internet Engineering Task Force (IETF). Considering that IEEE 802.15.4 only describes the lower communication layers standards (PHY and MAC), 6LoWPAN was introduced in order to institute a uniform standard of the network layer which would allow interoperability for different types of equipment. 6LoWPAN facilitates IPv6 connectivity over 802.15.4 devices (that are by default throughput and battery limited) by compressing the IPv6 packets. Details on the 802.15.4 standard can be found in paragraph 2.3.2.

2.4.1.1 *Networking*

The 6LoWPAN working group has decided to utilize IPv6 at the network layer in view of the demand for IP addressing and security, as well as the further development of IPv6. This has brought up a significant problem mainly arising from the fact that the payload length supported by MAC in IPv6 is much bigger than what is supported by 802.15.4 MAC layer. The solution to this was to introduce an adaptation layer between MAC and network layers in order to achieve header compression, fragmentation, reassembly, and mesh route forwarding. The reference model of 6LoWPAN protocol stack is shown in Figure 5.

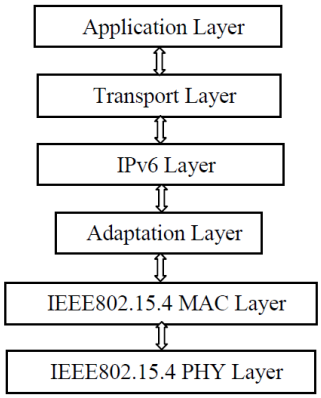

Figure 5: 6LoWPAN protocol stack reference model

Going into more details, below are the main networking features of 6LoWPAN:

- Adaptation Layer: The maximum frame size defined by 802.15.4 is 127 bytes, the maximum length of MAC head is 25 bytes, so the maximum length of the remaining MAC payload is 102 bytes. However, in IPv6, the maximum length of MAC payload is 1280 bytes, the IEEE802.15.4 frame cannot package the whole IPv6 data packet. Hence, it is necessary to introduce the adaptation layer at the layer below IP to achieve the functions of fragmentation and reassembly.

- Address Assignment & Management: 6LoWPAN has the capability of dynamic assignment of 16-bit short addresses. This gives the advantage of implementing hierarchical routing. Also, IPv6 supports stateless address auto-configuration which is suitable for the features of LR-WPAN equipment. Considering that LR-WPAN equipment may be largely and densely distributed in places where people are difficult to reach, having the ability of implementing stateless address auto-configuration gives an important advantage.

- Network Management: Network management technology is crucial to LRWPAN. Due to the large scale of network and the distribution of equipment in place, LR-WPAN should possess self-healing ability, and LR-WPAN management technology is required to be able to manage highly dense deployment equipment with a very low expense. 6LowPAN is apt to use SNMPv3 (Simple Network Management Protocol) in LR-WPAN to progress network management. Nevertheless, the original intention of SNMP is to manage IP-based Internet, so in order to apply SNMP to LR-WPAN, which has restricted hardware resource, further research and improvement are required, for example restrict data type, simplify radical code rule and so on.

2.4.1.2 *Power considerations*

6LoWPAN uses IEEE802.15.4 as its foundation, thus exploiting its advantages in terms of power usage and preservation. Having the ability to operate any device as host, it can facilitate very low consumption since devices can operate in sleep mode, waking up only periodically to check its parent device for connectivity. This allows for very low power consumption for the many devices in the WPAN network which are not required to operate as routers.

2.4.1.3 *Thread*

Thread is an IPv6-based, low-power mesh networking technology for IoT products, intended to be secure and future-proof. The Thread standard is created and managed by the "Thread Group".

The Thread Group was created in 2014 by a number of companies which intended to develop a complete standard which could be used commercially for home automation IoT products. Some of the biggest members of the Thread Group alliance are Nest (member of Alphabet/Google), Samsung, ARM, Qualcomm, Somfy, OSRAM, and Apple.

Thread is based on 6LoWPAN (which in turn uses IEEE 802.15.4) which is already been presented in paragraph 2.4.1 above.

Main characteristics of Thread include:

- Simple network installation, start up, and operation: The simple protocols for forming, joining, and maintaining Thread Networks allow systems to self-configure and fix routing problems as they occur.
- Secure: Devices do not join the Thread Network unless authorized, and all communications are encrypted and secure.
- Small and large networks: Home networks vary from a few to hundreds of devices, communicating seamlessly. The network layer is designed to optimize the network operation based on the expected use.
- Range: Typical devices in conjunction with mesh networking provide sufficient range to cover a normal home.
- No single point of failure: The stack is designed to provide secure and reliable operations even with the failure or loss of individual devices.
- Low power: Host devices can typically operate for several years on AA type batteries using suitable duty cycles.

2.4.1.4 *Networking*

The Thread protocol is based on 6LoWPAN, and it is further developed to become a commercially applicable protocol in the IoT industry and especially in home automation applications.

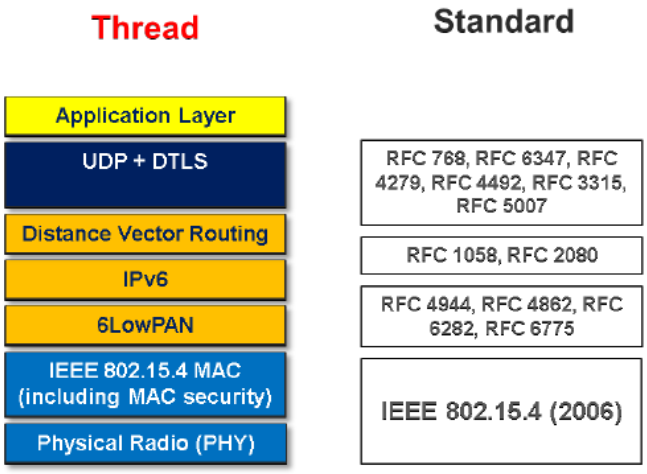

Figure 6: Thread protocol stack

2.4.1.5 *Device roles and types*

There are two types of forwarding roles in a Thread network:

- Router: a node that forwards packets for network devices, provides secure commissioning services for devices that are trying to access the network, and keeps its transceiver enabled at all times.
- End Device: a device that communicates primarily with a single router, does not forward packets for other devices, and can disable its transceiver to reduce power.

In addition, Thread nodes comprise a number of types:

- Full Thread Devices: they have their radio always on, subscribe to all-routers multicast address, and maintain IPv6 address mappings. They are split to three types:
  - Router
  - Router Eligible End Device (REED): can be promoted to a router
  - Full End Device (FED): cannot be promoted to a router
- Minimal Thread Devices: they do not subscribe to multicast traffic and they forward their traffic to their Parent. They are split to two types:
  - Minimal End Device (MED): their transceiver is always on thus they do not need to poll for messages from its parent
  - Sleepy End Device (SED): normally they are disabled, they wake up on occasion to poll for messages from its parent

Other roles and types are:

- Thread Leader: a Router that is responsible for managing the group of Routers in a Thread network. It is dynamically self-elected for fault tolerance and acts as an aggregator and distribution point for network configuration information.
- Border Router: a device that can exchange information between a Thread network and a non-Thread network. It also configures a Thread network for external connectivity.

Device number limits in a Thread network are:

Role	Limit
Leader	1
Router	32
End Device	511 per Router

Thread tries to keep the number of Routers between 16 and 23. If a REED attaches as an End Device and the number of Routers in the network is below 16, it automatically promotes itself to a Router.

2.4.1.6 *IPv6 addressing in Thread*

The scopes of IPv6 unicast addressing in a Thread network are divided into:

- Link-Local: all interfaces reachable by a single radio transmission. They have a prefix of fe80::/16
- Mesh-Local: all interfaces reachable within the same Thread network. They have a prefix of fd00::/8
- Global: all interfaces reachable from outside a Thread network

2.4.2 802.11 / WiFi

802.11-1997 was the first wireless networking standard in the family of wireless networking technologies, which came to complement the traditional wired networking. This modern alternative networking relies on wireless technology, rather that wired networking which relies on cables to connect digital devices together.

The wireless networking technologies include:

- WiFi
- Bluetooth
- 5G, 4G, 3G cellular internet and
- Wireless home automation standards, like ZigBee and Z- Ware.

The wireless networking technologies introduce the flexibility of using wireless technology, providing portability and freedom of movement. On the other hand, wireless technology imposes security concerns and risks.

The IEEE 802.11 standard applies to wireless LAN (Local Area Network), which is a computer network that spans a relatively small area. The services and protocols specified in IEEE 802 map to the lower two levels (Data Link and Physical) of the seven-layer Open Systems Interconnection (OSI) networking reference model. OSI, is a conceptual model that characterizes and standardizes the communication functions of a telecommunication or computing system without regard to its underlying internal structure and technology.

Therefore, the modern WLANs are based on IEEE 802.11 standards and are marketed under the wireless LANs.

2.5 Long-Range Communications

In the context of Internet of Things, when it comes to long-range wireless communications the dominating factor that needs to be considered is energy consumption. This is due to the inverse-square law of radio propagation that all electromagnetic waves follow. In particular, the law states that in free space the power density ρ of an electromagnetic wave is proportional to the inverse of the square of the distance r from a point source or $\rho \propto \frac{1}{r^2}$. In realistic settings, such as that of a smart building or an urban environment, power density may dissipate much faster, inversely proportionally to higher powers of distance.

Given the highly constraint nature of IoT devices (among others, in terms of computational resources and energy reserves), we identify a key *trade-off* when it comes to IoT long-range communications. On one hand, there lies the necessity for increased energy efficiency on the IoT devices; on the other hand, there lies the need to successfully and reliably achieve IoT communication over long distances. Managing to *fine-tune* the trade-off between these two competing goals has been fundamental in successfully developing large-scale IoT applications in the context of smart cities, smart agriculture, and similar wide area domains. In this effort a series of IoT wireless communication technologies has been developed – Figure 7 depicts a taxonomy of the main technologies currently available (2019).

The common characteristic of these IoT technologies is the fact that they require a low duty-cycle operation by the radio module and the microcontroller unit of the IoT devices, thus achieving great energy efficiency. However, the long-range technologies achieve this by operating at low, sub-GHz frequency spectra (typically in the range of 800Mhz – 900MHz in Europe), which in turn imposes a low baud-rate (typically in the range of few bits per second). Nevertheless, this still suffices in the grand majority of the IoT applications.

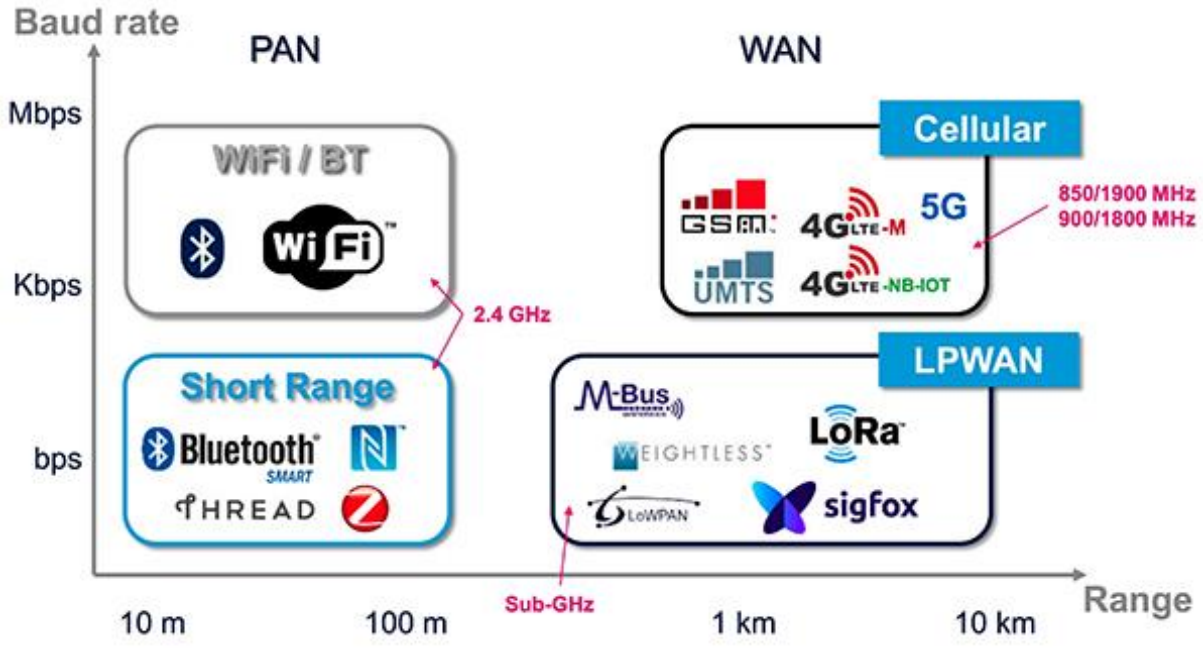

Figure 7: Taxonomy of IoT wireless communication technologies sorted by communication range (horizontal axis) and baudrate (vertical axis). Source: STMicroelectronics

2.5.1 LoRa

LoRa – standing for Long Range - is a Low Powered Wide Area Network technology developed by Semtech. LoRa uses license-free sub-gigahertz radio frequency bands - 868 MHz in Europe

- and enables very-long-range transmissions (more than 10 km in rural areas) with low power consumption. In order to deploy a LoRaWAN (i.e. a LoRa-based Wide Area Network), corresponding access points need to be deployed such that coverage is provided over the area of interest. A LoRa network server provides connectivity to other web-based systems (such as cloud-based databases). While the LoRa protocol is a closed, proprietary one, there are commercially available access points and end devices that anyone can procure in order to deploy and manage a LoRa network. Different users can make use of the same LoRa network server thus providing seamless roaming connectivity across different access points of different owners.

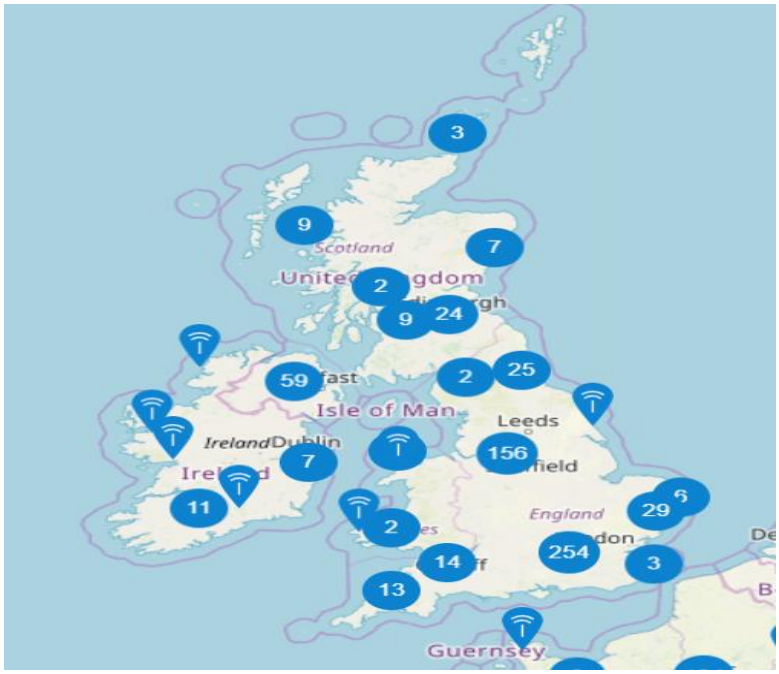

Figure 8: The Things Network – using the LoRaWAN - already demonstrates good coverage in the UK and it keeps growing. More information can be found at https://www.thethingsnetwork.org/map

Figure 9 depicts the reference architecture of a LoRaWAN. LoRa end-devices collect and broadcast IoT data which are picked up by one or more LoRa access points. Each access point has Internet connectivity (e.g. over Ethernet or cellular network) and forwards the data to the

LoRa network server. The server will resolve any duplicate messages (note that LoRa end-devices broadcast their data that can be picked up by multiple access points).

Globally, but particularly in Europe, there exists a strong community of LoRa users, and by following the aforementioned approach, good coverage has already been achieved.

Towards this direction, there exist several national initiatives, such as that by Digital Catapult [19] and The Things Network [20] in the UK. The initiative – named Things Connected [21] - operates a LoRa network server following an open-access policy for the public. This way, anyone can purchase and deploy a LoRa AP and connect to the network, effectively extending it. This is why LoRa has also been characterised as a community network. This open-access policy also helps make LoRaWAN a financially affordable choice since no additional costs are inferred for transferring data over the network. This also means that LoRa network operators and providers have more control over the data managed by the network. In terms of security, the LoRa protocol provides end-to-end data encryption while also offering several layers of network-level security.

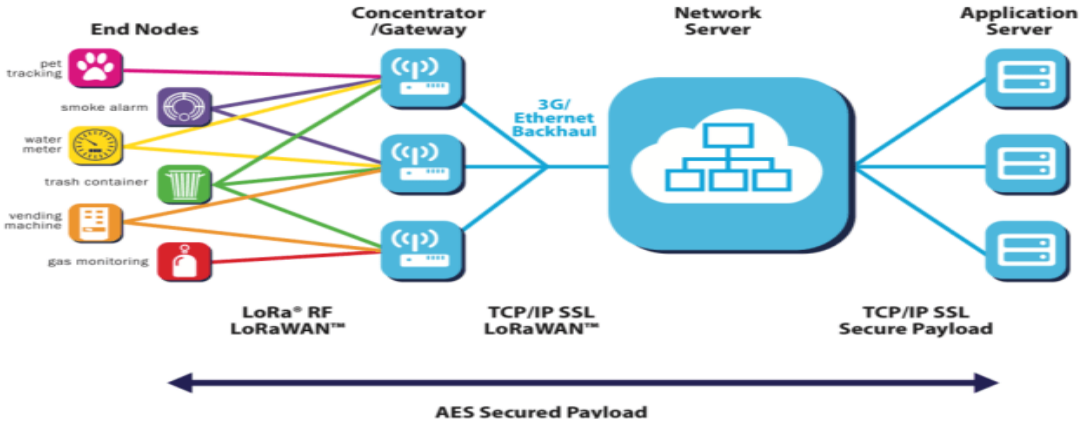

Figure 9: Reference architecture of a LoRaWAN network

2.5.2 SigFox

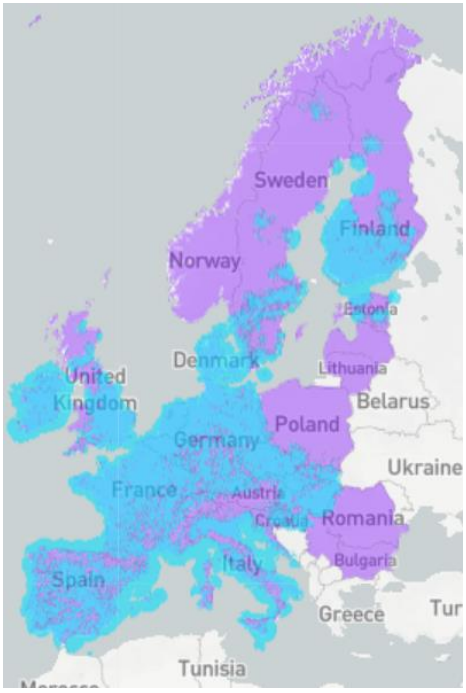

Figure 10: SigFox network coverage in Europe.

SigFox is another widespread Low Power Wide Area Network wireless technology developed by the homonymous French company. Similarly to LoRa, SigFox operates in sub-GHz frequencies, thus providing narrowband connectivity for IoT devices over long distances. However, SigFox follows a different architecture to LoRa which also dictates a different *modus operandi* and business model.

In particular, SigFox operates as a Network Provider (e.g. similar to a mobile network operator) who maintains the ownership of the access points it provides. This has the advantage of the SigFox network operating as a unified global network where no roaming services are needed. It also operates a single network server located in France. This has the implication of all network traffic being diverted to France before being forwarded to the customer's backend. Furthermore, if a given area is not already covered, it may take some time until the local provider deploys the needed access points. Finally, SigFox introduces upkeep costs related to the connectivity of the devices.

Regarding the network architecture, each end-device is associated with a single SigFox access point, thus creating a cellular-like network. Each device transmits its data to the associated at the time access point, which relays it to the SigFox network server over the Internet (e.g. via an Ethernet connection or over the cellular network). From there, data is forwarded to the back-end services of the corresponding application. An important limitation imposed by the SigFox protocol has to do with the way it manages its channel access. In particular, since the protocol operates in the un-licenced band, it needs to adhere to the restriction of occupying the wireless channel for less than 1% of the time; so, the protocol allows a maximum of 140 messages per day. While for several IoT applications this restriction does not pose significant

concerns, in some cases it needs to be accounted for. Finally, the maximum data payload per SigFox message is 12 bytes.

Figure 11: SigFox network architecture

2.5.3 Cellular LPWANs

The aforementioned LoRa and SigFox LPWAN protocols introduce new physical and MAC layers for IoT devices. As such, these are distinct new technologies that operate over their own specific network infrastructure. In an effort to leverage upon the already existing infrastructure, two protocols have been developed and are currently deployed across Europe; namely NB-IoT and LTE-M (also known as LTE-MTC). These protocols operate over the existing cellular infrastructure and therefore leverage upon good coverage, both indoors and outdoors. This also means the protocols operate on licensed spectra and therefore network services will be provided by mobile network operators, thus incurring corresponding operational costs. Both technologies are defined by the 3GPP in Release 13 and meet the requirements for low cost, low-power, and extended coverage. However, the two technologies demonstrate distinct characteristics, therefore making each one suitable for different types of applications.

NB-IoT (standing for narrowband IoT) introduces new physical layer signals and channels that are designed to meet the demanding requirement of extended coverage – rural and deep indoors – and ultra-low device complexity. Supported by all major mobile equipment, chipset, and module manufacturers, NB-IoT can co-exist with 2G, 3G, and 4G mobile networks. It also benefits from all the security and privacy features of mobile networks, such as support for user identity confidentiality, entity authentication, confidentiality, data integrity, and mobile equipment identification. NB-IoT has been developed to target low throughput devices and applications with limited dynamics and mobility. As such, it is perceived to have cost and coverage advantages over LTE-M, which supports a wide range of IoT applications, including content rich ones.

LTE-M (standing for Long Term Evolution for Machines), similarly to NB-IoT, is a cellular LPWAN technology that compared to NB-IoT supports comparatively higher data rates, mobility, and voice over the network, but it requires more bandwidth, introduces higher costs, and cannot be put into guard band frequency. Due to the fact that it supports much lower latency and higher network speeds, LTE-M is more suitable for time-critical applications, applications that make use of rich content or applications characterized by dynamics and mobility.

2.5.4 5G

Contrary to previous generations of cellular networks, such as 3G and 4G, 5G does not refer to a single wireless technology but to an ecosystem of technologies. The aim is to provide network services of significantly increased quality to end-users covering a highly diverse set of applications, ranging from massive IoT deployments, to autonomous vehicles, machine-to-machine communication, ultra-fast Internet connections on end-user devices, very dense network deployments, reliable tele-medicine applications, and other. Development of corresponding standards is driven by the following requirements:

- Up to 10Gbps data rate - > 10 to 100x improvement over 4G and 4.5G networks
- 1-millisecond end-to-end latency
- 1000x available bandwidth per unit area
- Up to 100x number of connected devices per unit area (compared to 4G LTE)
- 99.999% network availability
- 100% area coverage
- 90% reduction in network energy usage
- Up to 10-year battery life for low power IoT devices

Figure 12 provides a visual guide on the improvements on network services that 5G networks will deliver compared to existing 4G networks.

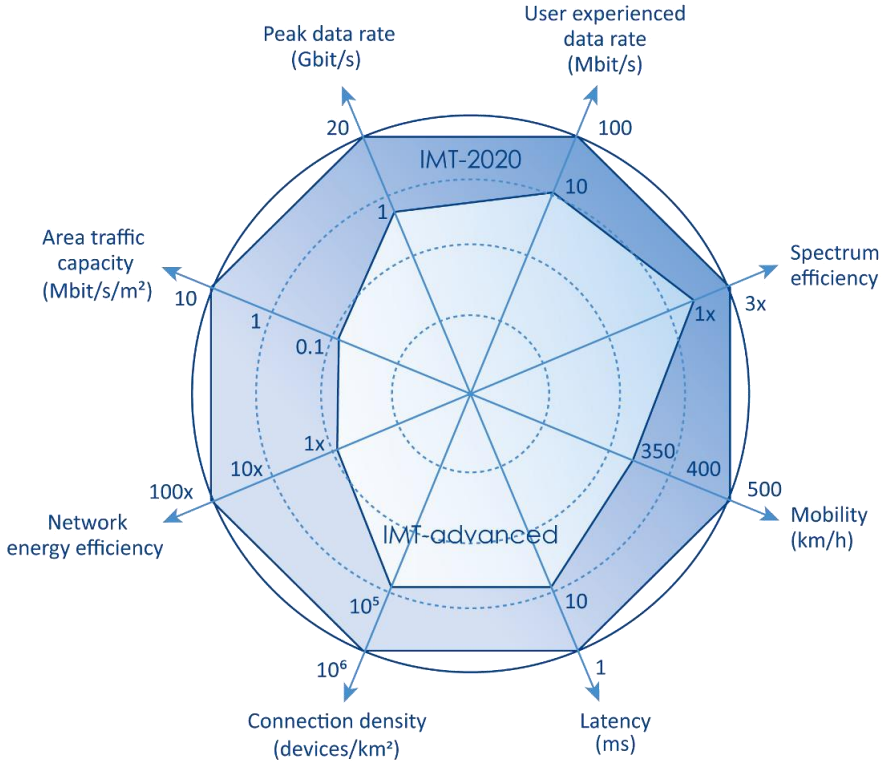

Figure 12: Comparison of key capabilities of 4G (IMT-Advanced) with 5G (IMT-2020) according to ITU-R M.2083

For 5G networks to achieve the specified requirements several technologies are employed, both at the wireless interface and the backhaul network. In particular, regarding the wireless interface, a variety of technologies will be employed, each one addressing a different segment of the 5G applications. Low power wireless technologies operating at sub-GHz frequencies will provide ultra-energy efficient connectivity for IoT devices. Frequencies in the areas of 1GHz

up to 3.4GHz will serve mobile applications such as video streaming and vehicular ad-hoc networks. Very high frequencies in the ranges of 30GHz to 60GHz – commonly referred to as mmWave technologies – will be used to provide fixed point-to-point wireless connectivity, for example among buildings.

The synergistic operation of these technologies will be orchestrated by recently introduced network management technologies deployed at the core and the edge of the network. Software Defined Network (SDN) and Network Function Virtualisation (NFV) software will allow the automated and agile management of core network resources, while corresponding software at the edge (such as in Multi-access Edge Computing – MEC) will manage cognitive and smart antennae.

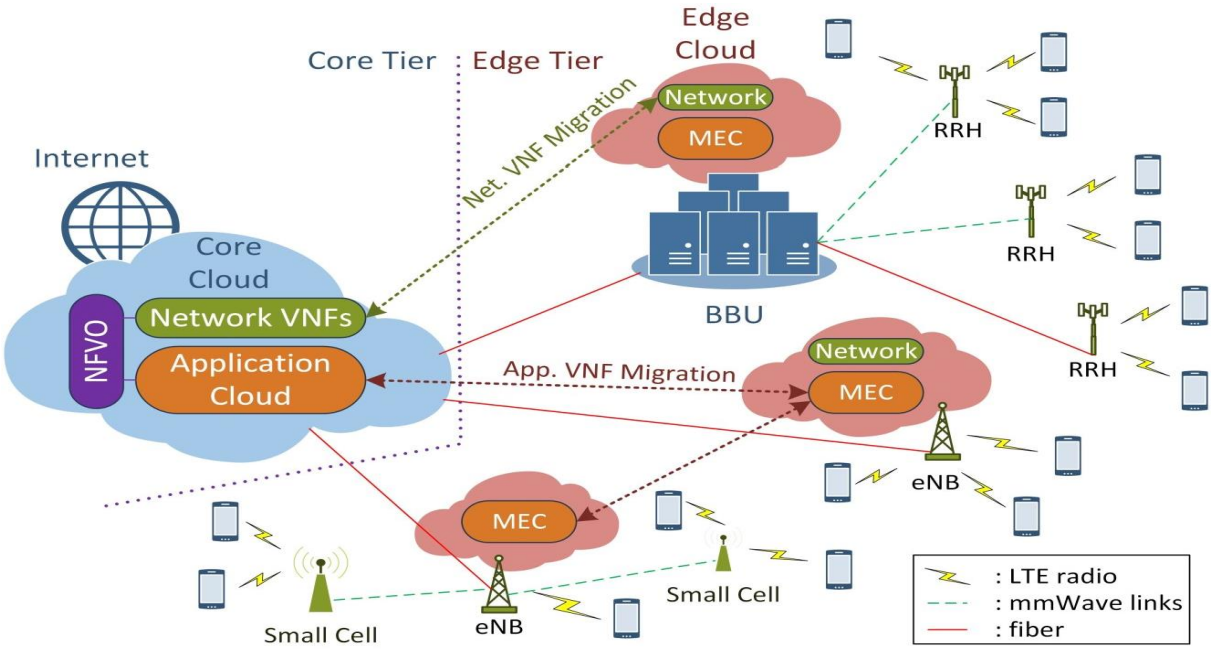

Figure 13: Indicative reference architecture of a 5G network.

2.6 Spectrum Sensing and Allocation

The radio spectrum is one of the most important resources for communications. Usually, radio spectrum is governed throughout the world by regulations and legislation has tended towards exclusivity of its use in geographic areas, allocating frequency bands for specific applications and assigning licenses to specific users or service providers. This fact has led to the shortage of wireless bandwidth for the emerging wireless services and applications. In addition, licensing of spectrum usually result to its underutilization. This status triggered the research for techniques that give the ability to unlicensed users to exploit unused licensed bands in an opportunistic way with the requirement that there is no interference with the licensed ones.

The results of this research is the Cognitive Radio approach which offers two prerequisite features for achieving opportunistic access: the functionality of spectrum sensing and the ability of the transceiver to change its configuration in runtime.

2.6.1 Cognitive Radio Principles

According to ITU [23] Cognitive Radio (CR) is “A radio system employing technology that allows the system to obtain knowledge of its operational and geographical environment, established

policies and its internal state; to dynamically and autonomously adjust its operational parameters and protocols according to its obtained knowledge in order to achieve predefined objectives; and to learn from the results obtained". A CR is a radio that can be dynamically configured to use wireless bandwidth in its vicinity with interference avoidance. Such a radio automatically detects available channels in wireless spectrum, and then accordingly changes its transceivers configuration allowing more concurrent wireless communications in a given spectrum band at one location [24].

CR is leveraged by Software Defined Radio (SDR), a new paradigm for realizing wireless communication systems. In a traditional radio system, the radio front end is composed by several subsystems (e.g. mixers, filters) that are implemented in hardware and tuned to operate with pre-defined settings (e.g. bandwidth, frequency). Thus a legacy radio transceiver is designed to operate for a specific service and it is very difficult or even impossible to be used it for a different one.

SDR came to bypass this disability. In an SDR transceiver the several sub-systems required are implemented in a computer via software. This way, not only a specific transceiver architecture is able to fine tuned to a specific service, but the whole transceiver architecture can be redefined even on run-time.

According to ITU [23], SDR is defined as "A radio transmitter and/or receiver employing a technology that allows the RF operating parameters including, but not limited to, frequency range, modulation type, or output power to be set or altered by software, excluding changes to operating parameters which occur during the normal pre-installed and predetermined operation of a radio according to a system specification or standard". SDR adopts a wireless digital transceiver architecture, where transceiver digitization (analog to digital and digital to analog conversion) occurs at some stage downstream the antenna (e.g. after some wideband filtering, amplification, frequency conversion). Reception of the signal is performed based on digital signal processing (DSP) approach. DSP offers great accuracy and flexibility in performing reception and transmission task that in a legacy radio are performed in tailored analog hardware. It is important to note that SDR differs from a digital radio in the fact that in a digital radio only the baseband processing is performed digitally.

An SDR radio typically consists of the following:

- Antenna: used to transmit /receive signals. SDRs typically employ several antennas in order to cover a wide range of the wireless spectrum.
- RF front-End: a set of analog RF circuits with the duty to amplify and convert the signal from/to the desired frequency band. Typically, the RF-front-end converts the used frequency band into an intermediate frequency band (IF) able to be digitally processed, however there are occasions that no frequency conversion occurs (e.g. when the used band is low enough for digital conversion and processing to be applied).
- Analog to Digital (A/D) and Digital-to-Analog (D/A) conversion: this is where the signal is converted to/from digital in order to be digitally processed. In the transmission path, the D/A converts the digitally generated waveform into an analog signal to be transmitted, whereas in the reception path, the A/D converts the received analog signal to digital in order to be digitally processed and demodulated.

- Digital front-end: used to perform further functions on the received/transmitted signal that were performed in the front-end in legacy systems (e.g. filtering, frequency conversions quadrature mixing), as well as sample rate conversions.
- Signal processing: responsible to perform baseband signal processing (similar to a digital radio) such as encoding, constellation mapping, modulation, scrabbling etc.

With regards to hardware implementation, the SRD approach requires a signal processing unit combined with a special hardware that implements the RF-front end and A/D-D/A functions. Several approaches are available [24] based in General Purpose Processors (e.g. an x86/64 CPUs), Graphics Processing Unit (GPUs), Digital Signal Processors (DSPs) and Field-Programmable Gate Arrays (FPGAs). Of course there are approaches that use combination of the above in a hybrid approach.

As mentioned above, CR is an all-intelligent radio that is able to utilize all the unused frequency spectrum to the best of the available resources. The major advantage of CR is that it can detect channels that are available from the spectrum and modify the parameters used for transmission so that the several unused frequencies can be used concurrently. To achieve this a CR follows the cognitive radio cycle shown in Figure 14 in its operation.

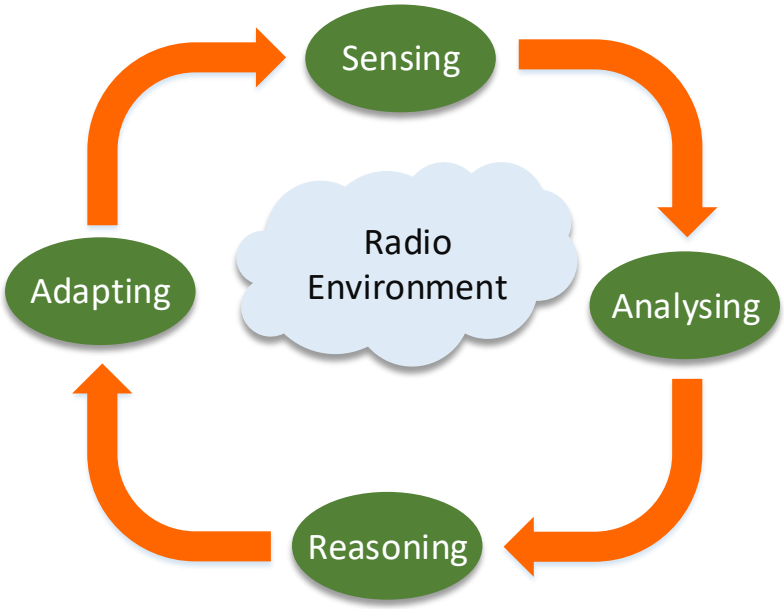

Figure 14: The Cognitive radio cycle [25]

During the sensing phase, the CR monitors the activity in the spectrum and detects white spaces that can be used for communication. This monitoring is is real time and should ensure that primary users will not be interfered. After the sensing phase, an analysis phase follows that has the task to recognise the white space with suitable frequency that offers the highest quality. Noise levels, losses, and error rates may be considered as the parameters to analyse in order to assess the quality of the frequency available. It is important to note that the frequency specifications are available for all the users (primary and secondary). After the analysis, a reasoning phase is following. This phase focusses in determining the best response strategy for the frequency allocation. The efficiency of the system is affected by the performance of this phase. Finally, an adaptation phase is reached where the CR utilises its SDR architecture to tune its radio to the selected band and optimise transmission/reception

configuration to the channels characteristics and of course eliminating interference for the PUs.

2.6.2 Spectrum Management Frameworks

CR networks impose unique challenges due to their coexistence with primary networks and diverse Quality of Service (QoS) requirements. Thus, new spectrum management functions are required for CR networks with the following critical design challenges:

- **Interference avoidance:** CR networks should avoid interference with primary networks.
- **QoS awareness:** To decide on an appropriate spectrum band, CR networks should support QoS-aware communication, considering the dynamic and heterogeneous spectrum environment.
- **Seamless communication:** CR networks should provide seamless communication regardless of the appearance of primary users

There are three major categories of the way CR access the spectrum [26]:

- **Underlay approach:** This approach mandates that concurrent non-cognitive and cognitive transmissions may occur only if the interference generated by the SUs at the PUs is below some acceptable threshold. The interference constraint for the PUs may be met by using multiple antennas to guide the SUs signals away from the PUs receivers, or by using a wide bandwidth over which the cognitive signal can be spread below the noise floor, then de-spread at the cognitive receiver.
- **Overlay approach:** the SUs use sophisticated signal processing and coding to maintain or improve the communication of Pus, while also obtaining some additional bandwidth for their own communication.
- **Interweave approach:** The 'interweave' paradigm is based on the idea of opportunistic communication, and was the original motivation for cognitive radio. This approach exploits temporary space-time-frequency voids to send information, and requires knowledge of the activity information of the PUs in the spectrum.

Several studies are trying to provide an efficient approach of managing spectrum in a dynamic and opportunistic way. In [27] for example, a spectrum management approach based on the belief vector concept is proposed. This approach characterizes and predicts the dynamics of the interference affecting a given radio environment and relies on a knowledge management entity that extracts the relevant knowledge from the radio environment. Authors in [28] propose a strategy based on a Partially Observable Markov Decision Process, whose target is to maximize a reward function that reflects the suitability of the available spectrum blocks to the application requirements. In [29] the authors propose dynamic spectrum allocation model based on the Cournot Game, which, according to them, improves the existing spectrum pricing function and introduces the SUs' competition factor. On this approach, the spectrum allocation behaviour of SUs is analysed by static and dynamic games. A preference-based truthful double auction for dynamic spectrum access approach is proposed in [30] where multiple heterogeneous spectrum bands are sold by the primary users and bought by the secondary users.

2.6.3 AI-based Spectrum Allocation

As noted above, awareness, reasoning, and learning are basic functionalities of a CR. Awareness refers to the process acquiring knowledge regarding the surrounding wireless environment. Reasoning refers to the strategy applied to act in a suitable manner in relation to the gained knowledge with the aim to maximize performance without interfering to PUs. Finally, learning is the knowledge accumulation (lessons learned) based on the results of the selection action (be reasoning) related to the previous knowledge. To such a process, powerful reasoning can improve the efficiency of learning.

Artificial Intelligence (AI) is increasingly applied in computer-related fields such as robotics, IoT, software defined networks, control systems, and simulation systems. Several tasks can be performed more effectively by AI than humans as AI is immune to emotions and fatigue. In recent years, after a long period of exploration and development, AI is becoming more efficient and sophisticated than ever. AI can leverage CR to achieve better efficiency in spectrum allocation. In [31] several AI approaches are explored as for their applicability to CR systems. The authors provide Table 8 summarizing their findings.

Table 8: AI possibilities for CR

Algorithm	Strengths	Limitations	Options
Artificial neural network (ANN)	Ability to describe a multitude of functions Conceptually easy to scale up Excellent for classification Can identify new patterns	Training may be slow depending on network size Possible over-training No underlying theory	Can use other learning techniques in the training phase Can be combined with RBS
Metaheuristic algorithms	Excellent for parameter optimization and learning involving relationship between parameter values Can use other learning techniques in the training phase	Formulation of rule space is difficult when learning or optimization is not restricted to parameter values	Can be used in conjunction with RBS Learning can also be used in the search process
Hidden Markov model (HMM)	Can model complicated statistical processes Good for classification Easily scalable Can predict based on experiences	Requires good training sequence Computationally complex	Based on previous knowledge, CBS and RBS can help HMM determine the observation duration for a specific application and overcome issues with new situations
Rule-based system (RBS)	Simple implementation	Tedious rule derivation process	Can be combined with CBS and OBS to

	Ability to tackle unforeseen situations Ability to include only relevant features while formulating a rule	Requires perfect domain knowledge which is not always available Not easily scalable	better deal with unfamiliar domain
Ontology-based system (OBS)	Ability to logically deduce Ability to understand the capabilities and characteristics of its own and others	Requires perfect domain knowledge to develop ontology Low efficiency for sophisticated ontology and ontology language	Can be combined with CBS and RBS to improve efficiency and robustness
Case-based system (CBS)	Close to human reasoning Can work in a chaotic situation with lots of variables Allows fast acquisition of knowledge Allows learning in the absence of domain knowledge	Relies solely on previous case Requires large case memory Might include irrelevant patterns	Can be combined with RBS and OBS to yield a more robust problem solving system that does not rely solely on experience

Several AI applications in CR have been explored in recent research. In [32] a Machine Learning (ML) approach is proposed to offer an adaptive channel assignment scheme based on fountain codes, in a bid to cope with the scarcity of existing cellular spectrum and the stringent requirements of ultra-reliable low-latency communication for 5G. In [33] the author highlight the applicability of learning based models for enhanced dynamic spectrum access in different traffic scenarios, enabling accurate prediction of primary OFF period, which is an important parameter for characterizing the spectrum utilization. [34] proposes a framework for the prediction-based efficient dynamic spectrum access, highlighting the importance of ML in the context of the CR for enhanced QoE of the SUs. In [35] the authors study the importance of real-time spectrum occupancy prediction in the context of CR for efficient DSA by using time-series models and ML techniques.

The conclusion from all these studies converged to the fact that AI and ML based techniques can increase the efficiency for CR deployments while minimising interference to the PUs.

3 IoT-Cloud Network

3.1 Background

IoT devices can generate massive amounts of data, which usually needs to be propagated to business applications residing in the backend for further processing. IoT devices and sensors could connect directly to the cloud backbone, but although this connectivity approach would greatly simplify the architecture of the IoT network, in practice it is rarely used. Modern networking devices host a plethora of functionalities such as service provisioning, switching between links, converting protocols, buffering data, tunneling, security etc. in order to meet service level agreements and deliver the required transmission quality. The constraints of IoT devices in terms of resources (e.g. power, hardware, software, administration features) prohibit them from performing these increasingly more demanding networking tasks. Consequently, specialized networking equipment in the form of network gateways needs to be deployed.

3.1.1 IoT Gateways

An IoT gateway can be either a specialized hardware device running a special purpose OS (e.g. Linux Ubuntu Core, Windows 10 IoT) and out-of-the-box IoT connectivity features, or a dedicated IoT connectivity software running on a commodity physical server. The former is usually engineered to withstand harsher environmental conditions (e.g. high/low temperatures, humidity, shock, vibration) and is therefore widely used in industrial settings and outdoors (e.g. Smart City).

Figure 15: IoT Hardware Gateway (Adlink Matrix MXE100i)

For the latter case, there are both commercial software solutions, e.g. by Bosch, Microsoft, Oracle, or Intel, open source solutions such as the Agile IoT, Eclipse Kura, or even plain software servers like Node.js optimized for M2M communication.

3.1.1.1 *Core capabilities*

At its most basic level, the IoT gateway will aggregate data from IoT devices operating within its WPAN, and send it over the WAN to a cloud backbone using widely used routing protocols, e.g. BGP, OSPF, or RIP. However, in addition to the WAN routing functionality which does not differ much from a standard router, in a real-world IoT setting, several provisions should be in place at the gateway. An indicative list thereof, is presented in Table 9 below.

Table 9: Required IoT Gateway capabilities in the project context

Provisioned Capability	Description	Applicable Use Cases
WAN Failover	Ability for the IoT gateway to switch between multiple WANs, (e.g. fixed to cellular) in case of communication failure	Network outages of a particular (cellular) provider; Mobile IoT gateways changing locations.
Dedicated service / emergency channel	A (low bandwidth) channel used to remotely inspect, control, and connect to the IoT Gateway and its devices in case all primary networks fail. It should not have any reliance on the primary networks infrastructure.	Large-scale disaster situations.
WPAN metrics reporting	The gateway collects and periodically reports various metrics and statistics pertaining to the WPAN operation and device health. Metrics can include data usage, reorganizations of IoT device mesh topologies, strength of radio signals, access control statistics (e.g. logins, attached clients) etc.	Resolve connectivity issues; manage usage costs; optimize network behaviour for different topology arrangements & scaling out
Secure VPN	Allows the gateway to establish a private communication channel to a remote network by tunnelling through a public network. This can be either done at the network layer for the entire connection (using IPSec), or at a higher layer for specific IP addresses via SSL/TLS (OpenVPN)	Exchanging data with a private, firewalled cloud network (e.g. a network handling sensitive user data)

3.1.1.2 *Edge computing*

The role of many gateways has moved beyond traditional routing, and often includes managing data for providing edge computing. The primary motivation for edge computing is avoiding unnecessary delays and bandwidth consumption by moving the processing of data as close as possible to the data source. Typical functions of edge computing are:

- Preprocessing and filtering of upstream data
- Managing events and applying rules
- Offline operation in case of a WAN outage
- Data storage and caching
- Analysis of security incidents

Another important driver is to process generated data of primarily local importance directly at its origin, thus avoiding complexity at the backend and helping the overall solution to scale better. This is particularly critical when important, data-driven decisions, relying on analysis of large amounts of data need to be made instantly. This can occur, for example, in cases where patterns need to be detected from a live video stream in order to alert users about

threats in the immediate environment. If this analysis would occur in the cloud instead of the edge, any WAN outage or temporary glitch would lead to delays which could have fatal consequences. Data analysis at the edge can involve:

- Anonymizing and denaturing of data, e.g. blurring of visual media
- Detecting complex patterns in data streams
- Training of machine learning algorithms to local usage patterns

3.1.2 Quality of Service

IoT gateways will usually serve multiple applications communicating to multiple IoT devices over a variety of networks. Given that, contention for the available network bandwidth by these applications is expected, and to avoid first-come first-serve behavior, traffic shaping and quality of service (QoS) mechanisms are used to ensure that each application can send and receive data according to its prescribed priority, and is not crowded out by bandwidth-intensive applications. Providing adequate QoS techniques at the router level is vital for project use cases that relate to citizen safety, handling of medical emergencies, or similar critical applications. That way, low-bandwidth but critical applications such as e.g. a heart-rate monitor will not be sidelined by a high-bandwidth but less urgent application such as e.g. a live video stream of a weather camera.

In addition to bandwidth, important parameters that QoS considers are delays, variation of delays (jitter), and packet loss. This is accomplished through defining channels with different quality guarantees and assigning the various traffic streams to each channel.

3.1.2.1 *Static and Dynamic Traffic Shaping*

We refer to traffic shaping as the process to reserve the necessary bandwidth to an application according to its requirements. Bandwidth can be allocated in advance to e.g. specific IPs, ports or packet types. This technique is referred to as Static Traffic Shaping, and although simple to implement at the gateway router, will usually result in sub-optimal bandwidth usage given that pre-allocated bandwidth which may be idle, cannot be used by applications other than the ones pre-assigned.

Contrary to the static technique, a dynamic traffic shaping mechanism will also guarantee bandwidth to pre-defined IPs, ports or packet types, but if and only if traffic is present for them. If traffic is not present, bandwidth will be freely allocated on a first-come-first-serve basis.

3.1.2.2 *Differential Services*

Further granularity to bandwidth allocation can be achieved by classifying and prioritizing different packets using a codification known as Differentiated Services Code Point (DSCP). DSCP is essentially a value in the packet header that determines the priority with which the packet should be handled by the routers in the network. The most commonly used priorities are:

- Default Forwarding (DF): The standard priority for traffic, will deliver packets in a best-effort manner
- Assured Forwarding (AF): Focuses on reliable traffic delivery. Furthermore, AF packets will be given higher priority than DF packets.

- Expedited Forwarding (EF): requests low loss, low latency and guaranteed bandwidth. This codification is typically reserved for the highest priority traffic.

For IoT-originating traffic, the IoT gateway can be configured to add the DSCP value, thus enabling classification according to the use case requirements. For scenarios where an urgent reply is desired, e.g. in citizen safety case, session-based DSCP can be used, which indicates to the remote router that the response should be marked with the same (high) priority as the request. However, although DSCP greatly facilitates the provision of QoS by only requiring intervention in the edge routers, there is no guarantee that routers between the IoT gateway and the cloud endpoint will honor the assigned priority. This is especially the case if packets need to traverse public networks, where intermediate routers must deal with a huge number of traffic streams. Furthermore, studies have also shown that there are still many routers which are still expecting/using the deprecated Type of Service (ToS) field for priority assignment [36], thereby permanently clearing the DSCP codification. As a consequence, careful network design and testing need to be performed to ensure that the desired traffic priority is actually delivered.

3.1.2.3 *Mean Opinion Score*

Especially for end-user targeted media traffic such as a video stream, VoIP, or general UI responsiveness, a measure of perceived quality known as the Mean Opinion Score (MOS) can be used in order to receive feedback about the actual end-user experience; this feedback can consequently be used to modify network routes, priorities, or even the application flow. Typically, perceived quality is rated according to the International Telecommunication Union's recommendations P.910 and P.800 five-level scale (1=Bad, 2=Poor, 3=Fair, 4=Good, 5=Excellent) [37][38]. The MOS is calculated as the arithmetic mean of all available ratings.

In addition to calculating the MOS using input provided by humans, recent approaches use algorithms to automatically determine the degradation between reference output and received signal without human feedback. Although not as effective as humans, these algorithms are considered practical for analyzing larger network topologies. The most notable algorithms for this case of MOS calculation are Perceptual Evaluation of Video Quality (PEVQ) for analyzing video, which is also included in the ITU P.910 recommendation, and Perceptual Objective Listening Quality Assessment (POLQA) for voice, which is the ITU P.863 recommendation [39].

3.1.3 Edge-To-Cloud Protocols

Behind an IoT system, there are constrained devices and protocols that handle all the communication in it. Constrained devices are equipped with sensor and communication capabilities to allow them to send data over the network. These constrained devices have limitations imposed by their limited resources, such as processing power, memory, and power consumption. In order to fit the needs for IoT systems, different types of protocols have been developed.

IoT devices are by nature heterogeneous systems, incorporating large number of hardware and software objects (e.g., cameras, microphones, traffic lights, actuators) that are not necessarily IP-enabled and may communicate via different technologies (Bluetooth, RFID, Zigbee, 802.11ah, 3G and 4G etc.) and carrier networks. Mobile phones can also be considered as IoT devices, with a variety of hardware and smart software objects. An IoT system should support the interconnection and networking of large numbers of such heterogeneous objects

(devices, software, and services) based on the use of lightweight standardized communication APIs and open software components so that it can overcome the limited capacity of wireless communications, particularly in dense urban environments.

3.1.3.1 *HTTP*

The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. HTTP has been in use by the World Wide Web global information initiative since 1990.

HTTP data rides above the TCP protocol, which guarantees reliability of delivery, and breaks down large data requests and responses into network-manageable chunks. TCP is a "connection" oriented protocol, which means when a client starts a dialogue with a server the TCP protocol will open a connection, over which the HTTP data will be reliably transferred, and when the dialogue is complete that connection should be closed.

The steps during a HTTP connection can be summarized as follows:

1. Client sends a SYN packet to the server.
2. Web server responds with SYN-ACK packet.
3. Client again sends an ACK packet, concluding a connection establishment. This is also commonly referred to as a 3-way handshake.
4. Client sends an HTTP request to the server asking for a resource.
5. Client waits for the server to respond to the request.
6. Webserver processes the request, finds the resource, and sends the response to client.
7. If no more resources are required by the client, it sends a FIN packet to close the TCP connection.

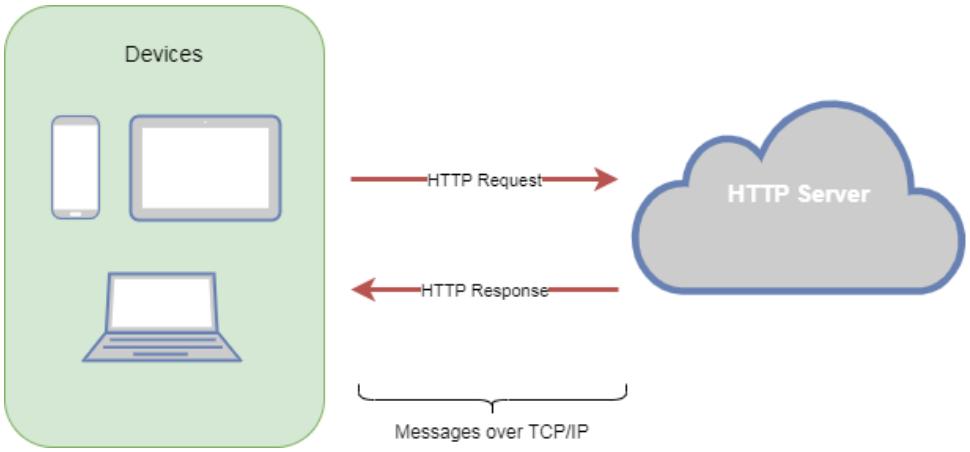

Figure 16. HTTP Client/device to server communication

HTTP is not very suitable for IOT, because of its following characteristics:

- One-to-one communication

- Uni-Directional
- Synchronous request-response
- Not designed for event-based communication
- High Power Consumption

3.1.3.2 *MQTT & MQTT-SN*

Probably the most widely adopted standard in the Industrial IoT to date, Message Queuing Telemetry Transport (MQTT) is a lightweight publication/subscription type (pub/sub) messaging protocol. Designed for battery-powered devices, MQTT's architecture is simple and lightweight, enabling low power consumption for the connected devices. Working on top of TCP/IP protocol, it has been especially designed for unreliable communication networks in order to respond to the problem of the growing number of small-sized cheap low-power objects that often appear in IoT networks.

MQTT is based on a subscriber, publisher, and broker model. Within the model, the publisher's task is to collect the data and send information to subscribers via the mediation layer (broker). The role of the broker, on the other hand, is to ensure security by cross-checking the authorization of publishers and subscribers. In MQTT are three levels of Quality of Service (QoS):

- QoS0 (At most once): The least reliable mode but also the fastest. The publication is sent but confirmation is not received.
- QoS1 (At least once): Ensures that the message is delivered at least once, but duplicates may be received.
- QoS2 (Exactly once): The most reliable mode while the most bandwidth-consuming. Duplicates are controlled to ensure that the message is delivered only once.

The client that publishes the message to the broker defines the QoS level of the message when it sends the message to the broker. The broker transmits this message to subscribing clients using the QoS level that each subscribing client defines during the subscription process.

Having found wide application in such IoT devices as electric meters, vehicles, detectors, and industrial or sanitary equipment, MQTT responds well to the following needs:

- Minimum bandwidth use
- Operation over wireless networks
- Low energy consumption
- Good reliability if necessary
- Little processing and memory resources

A variety of MQTT is MQTT-SN (MQTT for Sensor Networks). MQTT-SN is aimed at embedded devices on non-TCP/IP networks, whereas MQTT itself explicitly expects a TCP/IP stack.

Despite its characteristics, MQTT can be problematic for some very restrictive devices, due to the fact of the transmission of messages over TCP and managing long topic names. This is solved with the MQTT-SN variant that uses UDP and supports topic name indexing. However, despite its wide adoption, MQTT doesn't support a well-defined data representation and

device management structure model, which renders the implementation of its data management and device management capabilities entirely platform- or vendor-specific.

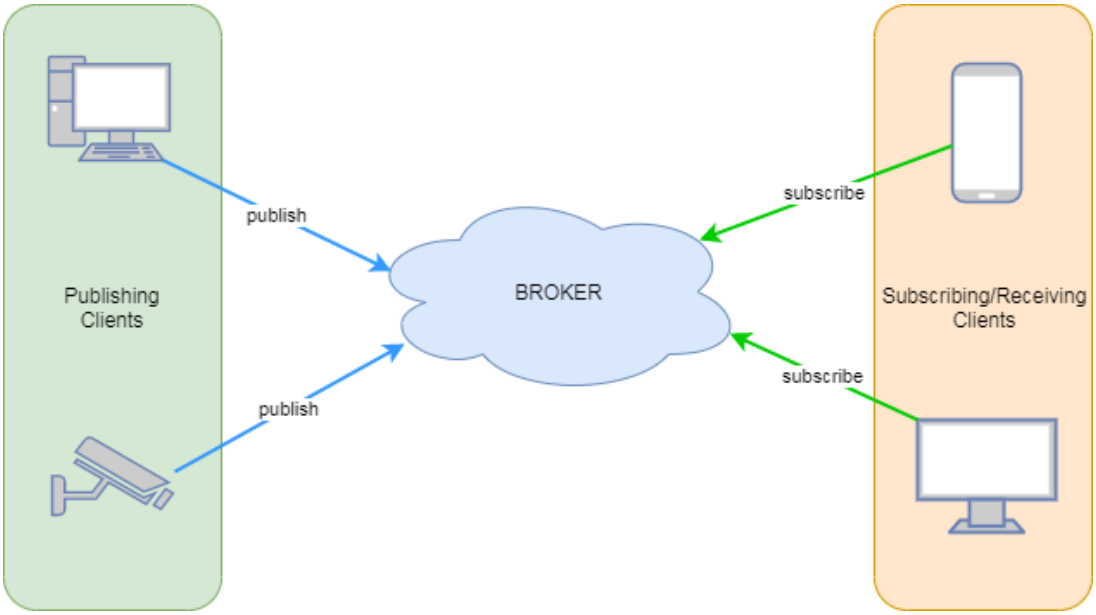

Figure 17 MQTT architecture

3.1.3.3 *CoAP*

While the existing Internet infrastructure is in principle available and usable for any IoT device, it often proves too heavy and power-consuming for most IoT use cases. Created by the IETF Constrained RESTful Environments working group and launched in 2013, the Constrained Application Protocol (CoAP) was designed to translate the HTTP model so that it could be used in restrictive device and network environments.

Designed to address the needs of HTTP-based IoT systems, CoAP relies on the User Datagram Protocol (UDP) for establishing secure communication between endpoints. By allowing for broadcasting and multicasting, UDP is able to transmit data to multiple hosts while retaining communication speed and low bandwidth usage, which makes it a good match for wireless networks typically employed in resource-constrained machine-to-machine (M2M) environments. Another thing that CoAP shares with HTTP is the RESTful architecture, which supports a request/response interaction model between application endpoints. Moreover, CoAP adopts the basic HTTP get, post, put, and delete methods, thanks to which ambiguity can be avoided at the time of interaction between clients.

The interaction model of CoAP is similar to the client/server model of HTTP. However, machine-to-machine interactions typically result in a CoAP implementation acting in both client and server roles. A CoAP request is equivalent to that of HTTP and is sent by a client to request an action (using a Method Code) on a resource (identified by a URI) on a server. The server then sends a response with a Response Code; this response may include a resource representation.

Unlike HTTP, CoAP deals with these interchanges asynchronously over a datagram-oriented transport such as UDP. This is done logically using a layer of messages that supports optional reliability (with exponential back-off). CoAP defines four types of messages: Confirmable, Non-

confirmable, Acknowledgement, Reset. Method Codes and Response Codes included in some of these messages make them carry requests or responses. The basic exchanges of the four types of messages are somewhat orthogonal to the request/response interactions; requests can be carried in Confirmable and Non-confirmable messages, and responses can be carried in these as well as piggybacked in Acknowledgement messages.

One could think of CoAP logically as using a two-layer approach, a CoAP messaging layer used to deal with UDP and the asynchronous nature of the interactions, and the request/response interactions using Method and Response Codes (see Figure 18). CoAP is however a single protocol, with messaging and request/response as just features of the CoAP header.

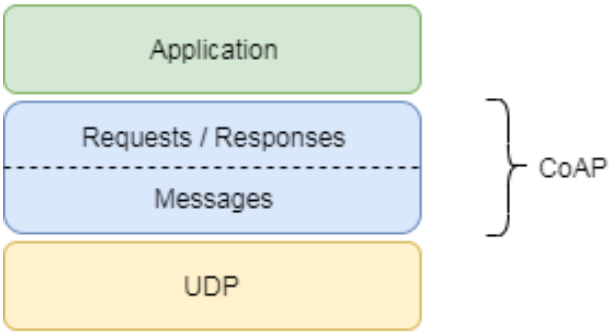

Figure 18: CoAP protocol layers

Summarizing, CoAP has the following main features:

- Web protocol fulfilling M2M requirements in constrained environments
- UDP [RFC0768] binding with optional reliability supporting unicast and multicast requests.
- Asynchronous message exchanges.
- Low header overhead and parsing complexity.
- URI and Content-type support.
- Simple proxy and caching capabilities.
- A stateless HTTP mapping, allowing proxies to be built providing access to CoAP resources via HTTP in a uniform way or for HTTP simple interfaces to be realized alternatively over CoAP.
- Security binding to Datagram Transport Layer Security (DTLS)

3.1.3.4 *AMQP*

AMQP is an open standard publish/subscribe type protocol originated in 2003, which has its roots in the financial services sector. While it has gained some ground within the information communication technology, its use is still quite limited in the IoT industry. The AMQP specification describes such features as message orientation, queuing, routing (including point-to-point and publish-and-subscribe), reliability and security. Probably the greatest benefit of AMQP is its robust communications model. Unlike MQTT, AMQP can guarantee complete transactions—which, although useful, is not always something that the IoT applications require.

Due to its heaviness, AMQP is not suitable for sensor devices with limited memory, power, or network bandwidth, yet for individual IoT use cases it may be the only protocol viable for end-

to-end application, including such examples as industrial heavy machinery or SCADA systems, where, as a rule, the devices and the network are considerably more capable.

AMQP is comprised of several layers. The lowest level defines an efficient, binary, peer-to-peer protocol for transporting messages between two processes over a network. Above this, the messaging layer defines an abstract message format, with concrete standard encoding. Every compliant AMQP process MUST be able to send and receive messages in this standard encoding.

3.2 Software Defined Networking

Software Defined Networking (SDN) [40] is an innovative architecture that has been widely adopted in the recent years for the deployment of mainly large-scale networks. Together with Network Function Virtualization (NFV) portray as the main technologies on which new generation network & infrastructure deployments will be based.

The concept of SDN is based on the decoupling of network control and forwarding functions. This allows for separate devices to be introduced in a network deployment which take the responsibility of these two main functions i.e. network control and packet forwarding. In addition, the introduction of APIs between the layers allows for different underlying networks to be controlled by the same controller.

Although concepts of programmable networking were discussed in the past, the separation of control and data planes was introduced back in 2004 by the Internet Engineering Task Force (IETF) which tried to standardize an open interface between control and data planes via the Forwarding and Control Element Separation (ForCES) [41] Working Group as well as introduce a logically centralized control of the network through the Routing Control Platform (RCP) and Soft-Router architectures. Nevertheless, the development of these concepts remained in academic level until 2010 when the first commercial deployments of SDN appeared. During this period Stanford University took the lead with the creation of OpenFlow [42] [43]as the first API for the communication between data and control planes and NOX [44]as the first network controller.

There is a large number of use cases for different types of deployments (i.e. carriers, service providers, data centers, cloud providers and enterprise networks) which could actually benefit from SDN. Features like bandwidth on demand, bandwidth calendaring, WAN optimization, network virtualization, network access control and network monitoring have proven to be very beneficial and have contributed to the recent wide adoption of SDN solutions.

A basic architecture of a Software Defined Network as shown in Figure 19 consists of three layers:

- The Infrastructure Layer (or Data Plane)
- The Control Layer (or Control Plane)
- The Application Layer

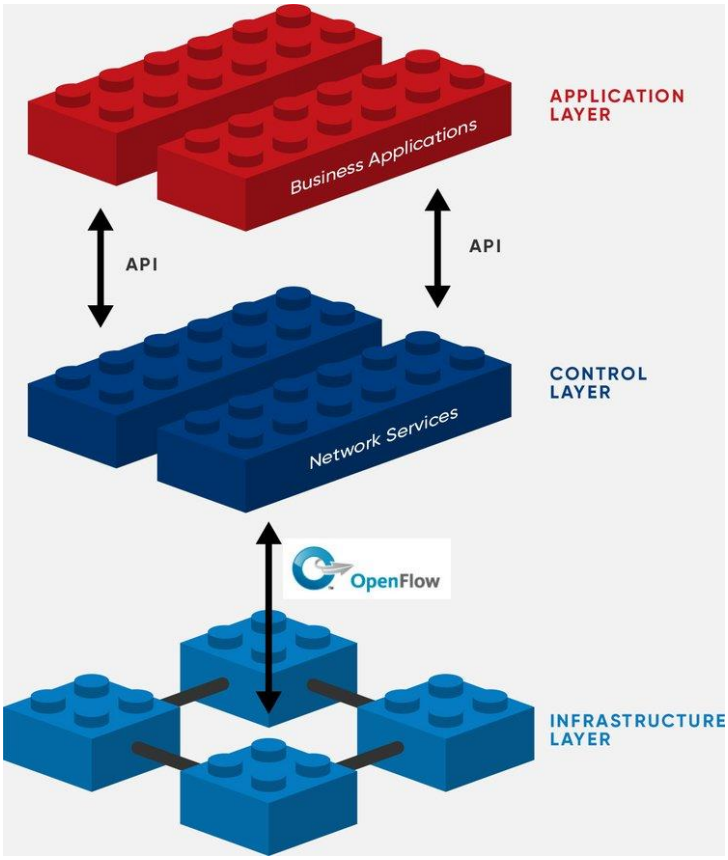

Figure 19: SDN architecture

3.2.1 Data Plane

The data plane in an SD Network consists of physical devices (mainly routers & switches) which forward packets based on instructions received by the network controller(s). These devices also implement the physical connections (Ethernet type) between them and to other external networks. The main difference between and SDN physical device and a traditional networking device is that in SDN the devices are “dumb” as they are controlled by a controller which sits in the upper layer (i.e. control plane).

Another interesting feature of SDN is the introduction of Virtual Data Plane. Apart from physical hardware forwarding devices, Virtual Data Plane are software-based emulators which can be installed on any physical machine and can provide great flexibility in the network deployment as they are programmable thus can be customized in any specific type of application or user requirements.

The communication between the data plane devices and the control plane is implemented via APIs. These connections are called southbound APIs and can be either open source or proprietary. The first and most well-known Southbound API is OpenFlow which was developed in Stanford and is promoted and maintained by the Open Networking Foundation (ONF). Other APIs are Network Configuration Protocol (NetConf) [45] which uses XML and also Lisp [46].

3.2.2 Control Plane

Control Plane in an SD Network consists of Network Controllers which act as the “brain” of the network deployment. They manage the network by managing the flow control to the

equipment below (via southbound APIs) as well as the applications and business logic above (via northbound APIs).

Since the inception of SDN many controllers have appeared in the market. Some of the most well-known are NOX (the first one), ONIX [47], POX [48], Beacon [49]and OpenDaylight [50].

The Control Plane can also be described as the connecting layer between the Data Plane and the Application Layer which handles the business logic, requests for needed resources and executes all user requirements. In addition, applications can also collect network information and build an abstracted view of the network that can be used for decision-making purposes. The communication between the Control Plane and Application Layer is implemented via Northbound APIs like RESTFul [51] API.

3.2.3 SDN in IOT deployments

As one can easily deduce from the above information, the concept and features of Software Defined Networking could potentially be a perfect fit to the IoT network deployment requirements. In an SDN-like IoT deployment the IoT devices can be implemented as the SDN data plane while at the application layer one can have an overall view of this network at any time thus provision any changes that facilitate near real-time needs at the data plane level. The SDN controller can then provide programmability and flexible management of the data plane flow control while at the same time issues like resource management, spectral efficiency, network management and transmission control at the data plane level can also be facilitated at the control and/or application layers.

Existing studies on possible SDN implementations for IoT mostly tackle wireless networks where the need for flexibility and programmability is more prominent.

A paper from Tayyaba et al [52] presents and reviews a list of existing IoT solutions based on SDN. There are a number of proposals which discuss SDN based cellular networks (mostly 4G LTE and 5G), SDN based IoT management frameworks that tackle the management of wireless sensor devices as well as resources allocation and configuration, and SDN based security frameworks. A basic conclusion from this review and comparison is that the combination of IoT and SDN is still in an immature stage although IoT and SDN are separately proceeding in a good pace thus providing solutions that are already adopted by the market. A comprehensive IoT-SDN architecture and framework which incorporates all aspects and can be considered as a complete and concrete solution does not exist to date although there are a number of efforts which have a high level of completion like SoftRAN [53], SoftAir [54], SDN-WISE [55], SDIoT [56] etc.

3.2.4 Operational Aspects

3.2.4.1 *Service Chaining*

Service chaining [57] refers to a feature that has been introduced in SDN where the administrator of the network can design and program sets of services inside the software defined network that can apply to different types of traffic flows.

In order to better understand this concept, we need to define what we mean by services. It is standard procedure that in traditional networks various types of equipment are installed in order to facilitate different functions. This equipment can be firewalls, WAN optimizers, load balancers etc. Each one of this equipment can support various functions like deep packet inspection, NAT, traffic control, traffic compression etc. In an SDN environment all of these

functions can be considered as services, they can be pulled out of the traditional network equipment and they can be implemented in virtual environments thus offered as a service.

Another issue that we need to have in mind is that in traditional networks all packets will have to travel through all the devices that the network owner has installed. However, the reality is that not every packet has to travel through all of the devices that realize a network since not every service has to apply on all of the packets that travel through it. This is where service chaining becomes important and very useful in an SDN environment. With service chaining the network administrator is in a position to create different service groups i.e. chains of functions that are available in the software defined network. The network controller could then apply these chains to different traffic flows according to their source, destination or type of traffic. This way many procedures that in traditional networks have been implemented manually can now be automated.

3.2.4.2 *Dynamic Load Management*

Dynamic Load Management [58] in a Software Defined Network provides the capability of even distribution and dynamic management of the traffic load in the data plane. By optimizing the available resources, the network administrator can increase throughputs, avoid congestion and decrease response times while at the same time expand the network topology dynamically according to the client needs. Using SDN, workflows can be changed dynamically to accommodate workloads more efficiently. SDN controllers utilize various algorithms which optimize the flows during transmission. The basis of these decisions are the load statistics from each data plane device and the priority of each traffic flow.

3.2.4.3 *Bandwidth Calendaring*

Bandwidth Calendaring [59] is a service in a Software Defined Networking environment that exploits knowledge about future traffic in order to handle the arising demands over the network and manage the network resources utilization accordingly.

Currently networks need to be overprovisioned in order to be able to accommodate expected peak traffic thus avoid congestion. With Bandwidth Calendaring the network administrator has the ability to dedicate network resources in specific time windows that are suited to accommodate large amount of traffic flows.

3.3 IoT Network Function Virtualization

The traditional approach in networking has relied on vendor-specific special-purpose network nodes, where hardware and software are tightly coupled. Thereby, the configuration of those nodes is rather costly and leads to a rather rigid network. However, this traditional approach hardly holds nowadays. There are several reasons for this. First, the number of devices requiring network connectivity and the data rate have increased dramatically, mostly due to a huge number of IoT devices and mobile terminals. Second, the appearance of IoT demands services with dynamic and heterogeneous QoS requirements. In this scenario, the traditional networking approach, based on vendor specific special-purpose nodes, leads to dramatic increases in Capital (CAPEX) and Operational (OPEX) Expenditure. These issues are circumvented thanks to a new networking approach based on NFV.

Network functions virtualization (NFV) refers to the process of separating network functions from hardware to create a virtualized network that can run on commodity hardware, allowing networks to be more pliable and more cost-effective. At the core of NFV are virtual network

functions (VNFs) that handle specific network functions, like firewalls or load balancing. Individual VNFs can be connected or combined as building blocks to create a fully virtualized environment. VNFs run on virtual machines (VMs) on top of the hardware networking infrastructure. There can be multiple VMs on one hardware box using all of the box's resources.

Network virtualization and VNFs development result from service providers working to speed up the deployment of new network services while reducing operating costs and capital expenses. IT virtualization technologies appealed to these providers to achieve those goals. On its face, virtualizing a network means reducing the scale, diversity, and cost of the hardware to only what is necessary; and using software for network functions, so if business needs change, providers can easily update the software instead of the whole system's hardware. The problems of the hardware network go beyond the cost of buying a new box, which once was an expensive proprietary product. The physical act of upgrading each piece of hardware was an immense drain on resources. Truck rolls, where hardware was delivered to the data center and people were paid to physically install new hardware, racked up costs and took up a lot of time. When virtualizing networks became feasible, providers turned to them to cut out additional costs. The shift from several pieces of hardware, each performing their single function, to a single piece of hardware with several VMs within it, each performing the actions of VNFs was necessary and a new standard was needed to make the transition easy. Those service providers founded the European Telecommunications Standards Institute (ETSI) in 1988, and in November 2012 they created a working group for NFV called the ETSI Industry Specification Group for NFV (ETSI ISG NFV). Within a year, ETSI published papers on NFV use cases and basic requirements. In 2014, they released a paper defining NFV architecture [60]. In the paper, ETSI heavily features VNFs as the foot soldiers of NFV.

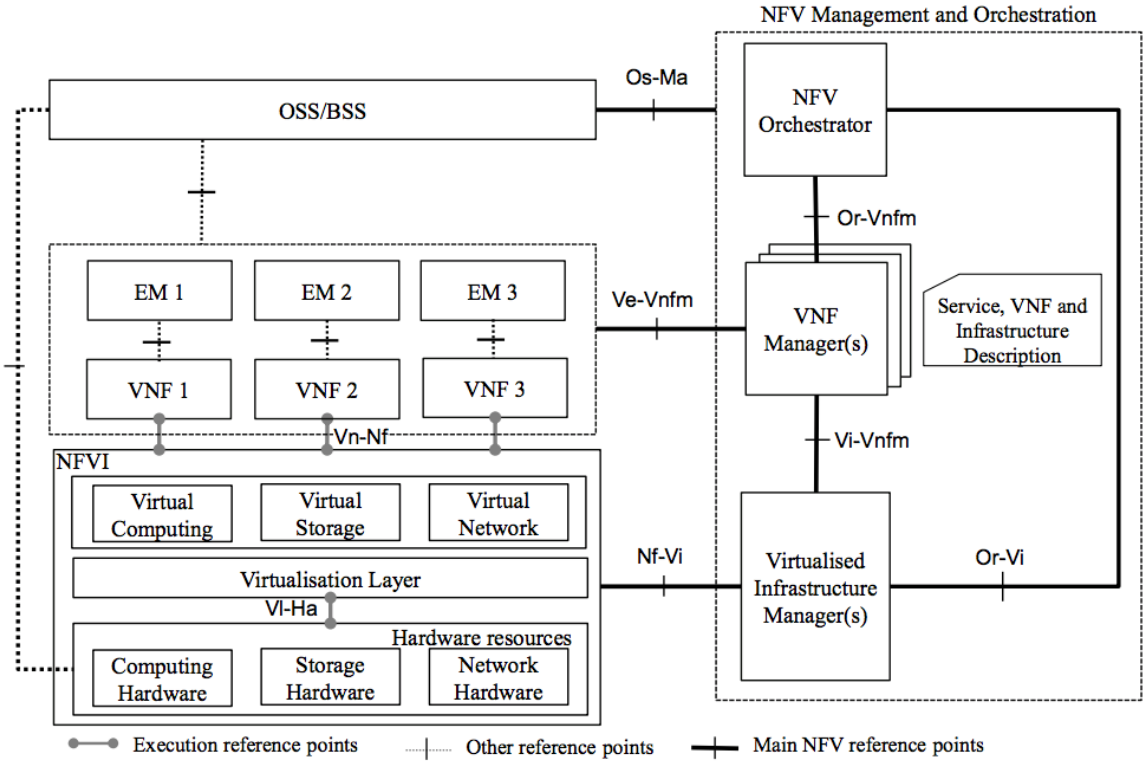

Figure 20 ETSI NFV Virtualization Architecture

3.3.1 Physical Layer

As aforementioned, the VNF architecture is used in order to minimize the hardware usage. Bari et al. [61] propose a model to optimize the VNF placement problem in ISP and enterprise networks while minimizing operational costs, mainly node and link resource utilization. As SLA constraint, they only consider propagation delay and do not include processing delay at each node. In the same context, Authors in [62] propose a VNF chaining and placement model that optimize network level (i.e., link utilization) and NFVI-level (i.e., allocated computing resources) performance metrics. For this, they consider the latency bounds at both the VNF node and the end-to-end levels. To determine the VNF forwarding latency metric, they only consider traffic load and do not consider utilization level of the physical host (i.e., the number of VMs placed on the NFVI node) which can dramatically increase the VNF latency due to the virtualization overhead and resource sharing of the same physical host. Moreover, both of [60] and [62] do not consider the latency between VNFs and end-users and the fact that some VNFs may require to be placed in the proximity of end-users. In [63], authors address the placement of virtual mobile core network functions (i.e., S-GW, PDN-GW, MME and HSS) excluding VNFs on the radio access network. Their optimization target is to minimize the cost of occupied link and node resources while taking as constraints VNF requirements in terms of bandwidth, processing and storage resources. However, they do not consider latency constraint on the VNF nodes and the end-to-end network. In the same context of mobile networks, Taleb et al. [64] propose algorithms to place VNFs of both PDN-GWs and S-GWs on a given topology of distributed datacenters. In [65], authors investigate the VNF placement problem in the radio access network (RAN) domain which can include functions such as load-balancing, firewall, and virtual radio nodes. Their objective is to minimize the cost of mapping virtual functions to substrate network (nodes and links) while satisfying VNF requirements in terms of CPU, memory, storage, radio, and bandwidth resources.

3.3.2 Virtualization Layer

The virtualization layer decouples the VNF software from the underlying hardware by abstracting the hardware resources available. The architectural view of NFV infrastructure is depicted in Figure 17. The virtualization layer in the middle ensures that VNFs are decoupled from hardware resources and therefore the software can be deployed on different physical infrastructure resources. Typically, this type of functionality is provided in computing and storage resources using Hypervisors.

The VNF architecture does not make restrictions on the layers of virtualizations. A VNF can have direct access to a network card for example for better performance. The use of the Hypervisors is the typical solution but not the only one. In cases where hypervisor support is not supported VNFs can run as application on the operating system. Moreover, when network virtualization is used the network hardware is abstracted to the application layer by using several techniques. Some of them are Virtual Local Area Network (VLAN), Virtual Private LAN Service (VPLS), Network Virtualization using Generic Routing Encapsulation (NVGRE).

3.3.3 Application layer

Virtualized network functions or VNFs are the software realisations of the various network functions that can be deployed on a NFV infrastructure and are needed in order to enable the network to operate.

In this way, a VNF handles a specific network function that runs on one or more virtual machines on top of the hardware networking infrastructure. The individual VNFs can be

considered to be building blocks and they can be connected or combined together, providing all the capabilities required to provide a complete networking communication service.

Examples of various virtual network functions can be found within all areas of a telecommunications network, and they can include:

- Switching: BNG, CG-NAT, routers.
- Tunnelling gateway elements: IPSec/SSL VPN gateways.
- Traffic analysis: DPI, QoE measurement.
- Signalling: SBCs, IMS.
- Application-level optimisation: CDNs, load Balancers.
- Home routers and set top boxes.
- Mobile network nodes: HLR/HSS, MME, SGSN, GGSN/PDN-GW, RNC.
- Network-wide functions: AAA servers' policy control, charging platforms.
- Security functions: firewalls, intrusion detection systems, virus scanners, spam protection.

In this way, it can be seen that a huge number of VNFs can be run on a network using network functions virtualization (NFV.

4 Location sensing in urban environments

The plethora of IoT devices, protocols, and networks already mentioned in the previous sections can be, and they actually are, used for geo-location purposes in urban environments. This is usually realised along with, or instead of, the widely used global navigation satellite systems (GNSS), like GPS and Galileo. Especially for indoors environments, where GNSS is practically inapplicable, IoT solutions are obvious candidates for location sensing and tracking of both people and things. Numerous approaches have been put forward for such purposes, using some of the technologies already reviewed in the previous sections.

Nevertheless, such solutions require installation and maintenance of an adequate infrastructure, which may limit their applicability. For the purposes of the IDEAL-CITIES project, we will seek to exploit an approach that largely avoids the need for such an infrastructure, namely *visual localization*, appropriated for use in a general IoT and Smart City context, such as the IDEAL-CITIES one.

Visual localization builds directly upon the recent vast and rapid progress in computer vision tasks, fuelled mainly by advances in deep neural networks (deep learning), now generally referred to as “Artificial Intelligence (AI)”; although we consider the term inappropriate and possibly misleading, we will use it in the present document, in conjunction with its general use in similar contexts and discussions (see also Section 2.6.3).

The recent so-called revolution in AI for computer vision tasks is considered to have begun in 2012, when a deep convolutional neural network improved the error rate of the ImageNet challenge by more than 10% compared to the second-best solution [82]. The relevant techniques and algorithms found quickly their way into visual localisation tasks [83].

Roughly speaking, the general idea behind visual localization using such AI techniques is cast as an image retrieval from a database task: a query image is used to visually search through a geotagged image database and return the closest image to the query one, along with its location [84]. The novelty introduced by the AI algorithms in the process (which in principle may sound like a classic database retrieval problem) is the effective and efficient feature extraction from the relevant images, which permits the whole process to be performed with very high accuracy in real-time using only modest computer hardware available at the edge.

The subject approach avoids the deployment and maintenance issues of a dedicated IoT infrastructure, effectively replacing it with a centrally administered image database (e.g. at City Council level) in the cloud and the use of commodity smartphones at the user level. Such databases can in principle be easily created and maintained up to date using crowdsourcing approaches. Input data at the edge are provided by vision sensors (Section 2.1.2), and the networking between devices and the cloud is typically over the 802.11 variants reviewed in Section 2.4.2 (WiFi, 3G/4G).

4.1 Urban outdoor environments

There has been significant work lately on visual localization approaches, with the majority of them being about urban outdoor environments. A demo application very close to what we aim for with the IDEAL-CITIES use case #1 is available for a limited area around the city center of Cambridge, UK [85]:

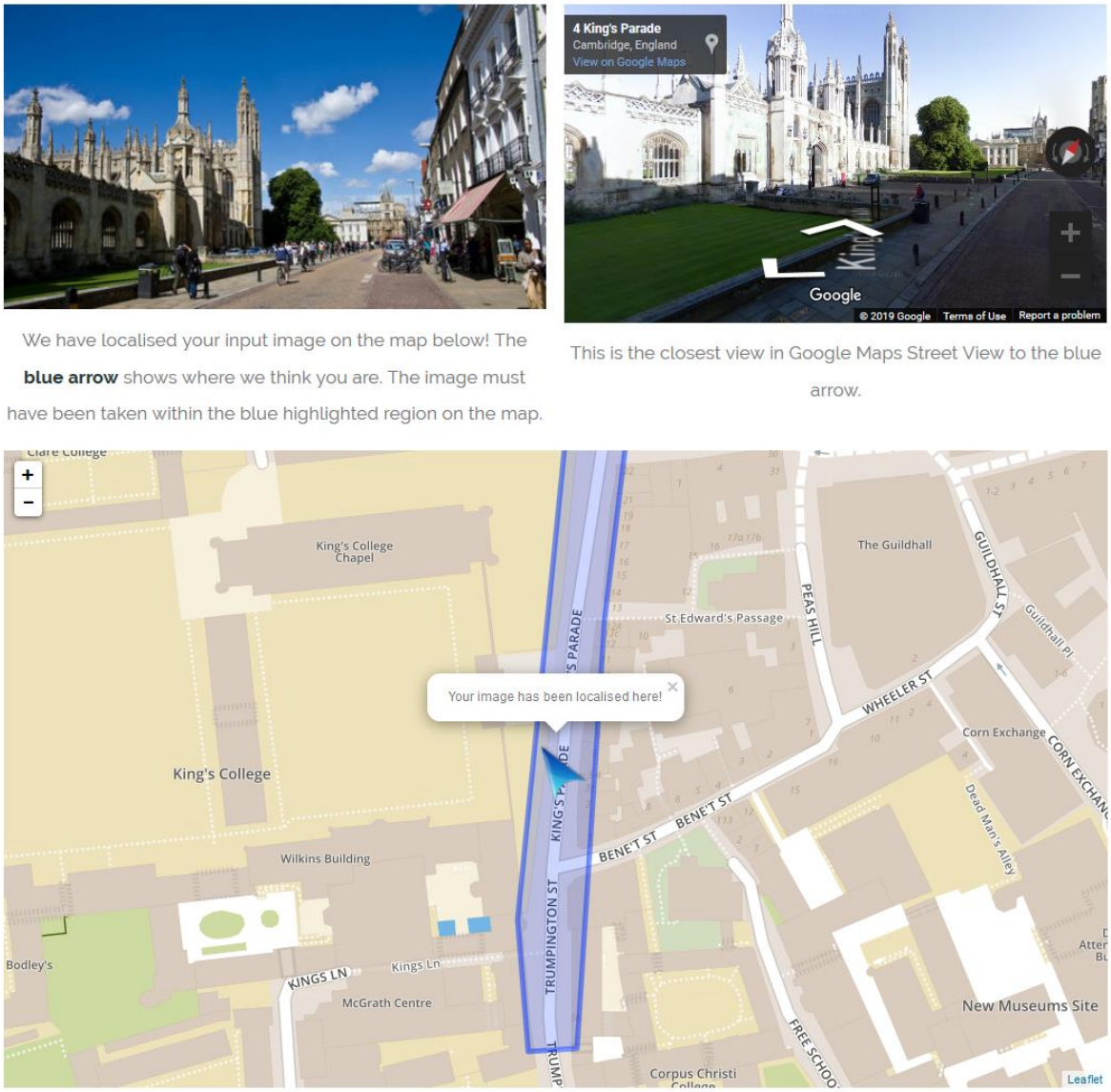

Figure 21: Query image (upper left) and inferred localization (bottom map) in central Cambridge, UK [85]

As noticed in [85], the process takes under 2 msec, and is more accurate than GPS, with the additional advantages over GPS that it includes orientation and it can also operate indoors.

A similar approach, but using bird's eye view reference images instead of street-level ones, is presented in [86], while an approach specifically targeted for live on-road localization of vehicles, namely Vehicle Localization by Aggregating Semantic Edges (VLASE), is presented in [87], with a demo video in [88].

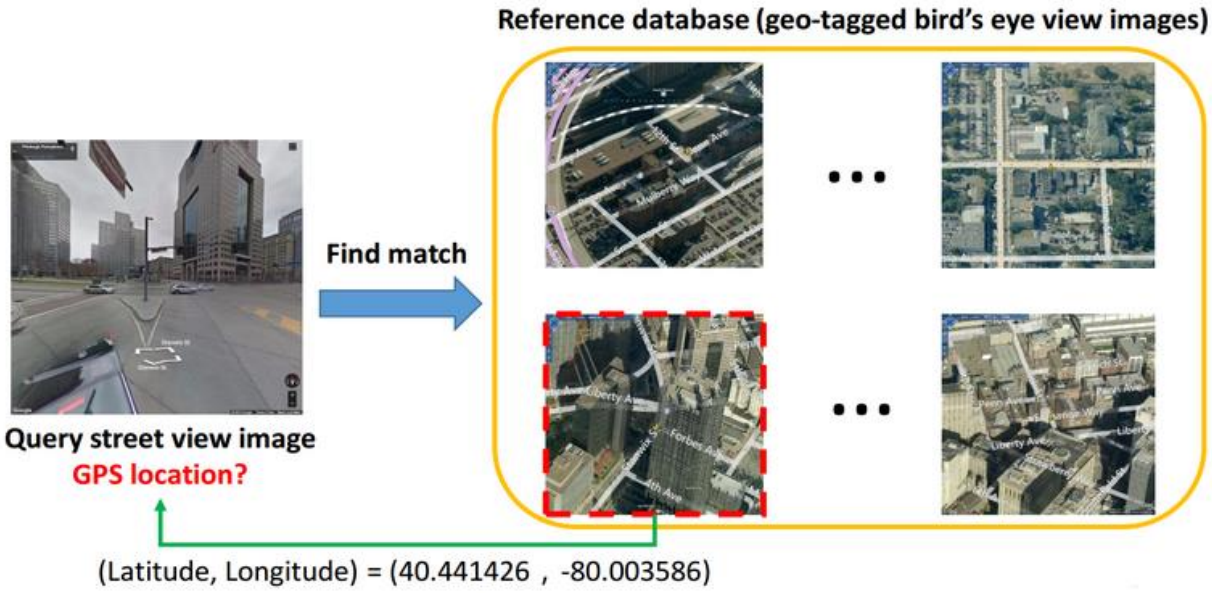

Figure 22: *Querying street-level images from a reference database of bird's eye view ones [86]*

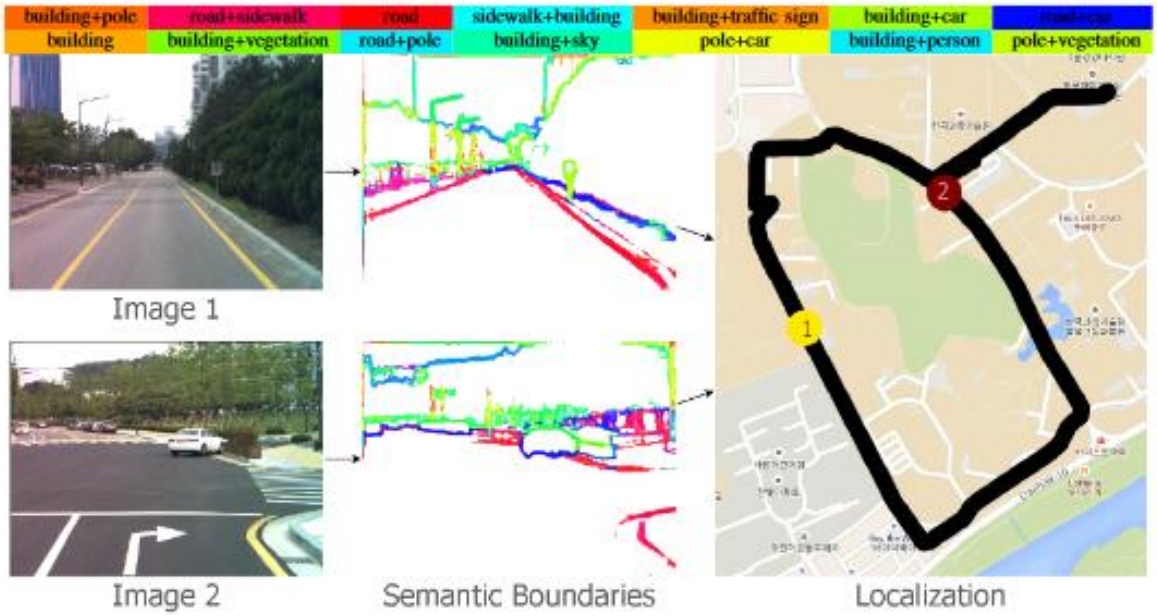

Figure 23: *Illustration of VLASE from [87]. Given images (left) from a vehicle, semantic edge features (middle) are extracted. Different colours indicate different combinations of object classes. The extracted semantic features are compared to the features from geo-tagged images in a database to estimate the location. In this example, the red and yellow circles on the map (right) indicate the locations of the two given images.*

All cases above are based upon the same principles mentioned in the previous sub-section, i.e. using deep neural networks for extracting features and perform image comparison between query images and stored ones. There exist many more approaches similar to the ones mentioned – a detailed survey is beyond the scope of the present document, and it would arguably be obsolete in a few months time, since this is a highly active application field, with new approaches reported on a very frequent basis. For our purposes here, it suffices to

register this category as a relatively new player in the urban localization field with promising potential and minimal infrastructure requirements.

4.2 Indoor environments

Although the reported demo applications for indoor environments are much fewer than the ones for outdoors, approaches such as the ones outlined in the previous sub-section are directly applicable to indoor settings, too. In fact, as often mentioned explicitly [85], this potential is understandably considered as a main advantage of this kind of methods, since GNSS approaches are by default not applicable here. Additionally, and if a relevant infrastructure is already in place for outdoors environments, extension to indoors ones is as simple as enriching an existing image database with new geotagged images, a task which does not require installation or maintenance of additional hardware, and is arguably appropriate for crowdsourcing solutions.

4.3 Related areas

A closely related area to visual localization is that of landmark detection and landmark-based localization, which has been also proposed in the past for assisting visually impaired people [89]. A particularly interesting aspect of this line of research is the increasing availability of relevant large datasets, such as the Accurate Landmark Positioning at City Scales (ALPS) [90] and the Google Landmarks datasets [91][92]. Although these datasets were not made specifically for visual localization of individuals, they may provide useful ground for experimenting with the relevant AI techniques involved.

5 Conclusion

It is apparent that there is a plethora of available options for an IoT infrastructure: some are targeted for long-range communications, while others are designed for short-range interactions; there exist both proprietary protocols and open ones; there are approaches suitable for a relatively large volume of data exchanged, and others designed to take advantage of situations where this volume is by definition small; some approaches exploit the already present connectivity backbone provided by cellular networks, while others come with their own dedicated one. A common concern to almost all of them is the power management of the employed devices; for options including wireless components, efficient spectrum allocation is an additional constraint.

All of these options, briefly reviewed in the present document, are in principle suitable for use in a Smart City context. This large number of options permits a great flexibility in designing solutions and customizing the provided services according to the context and the desired specifications. Specifically for location sensing, we have hinted at a relatively new approach, namely visual localization, which is able to provide high accuracy localization in both outdoor and indoor environments, thus overcoming the standard outdoors-only limitation of GNSS systems, like GPS and Galileo, while at the same time avoiding the necessity of a dedicated infrastructure beyond a cloud database.

6 References

[1] Rozsa, V., Denisczwicz, M., Dutra, M., Ghodous, P., Ferreira da Silva, C., Moayeri, N., … Figay, N. (2016). An application domain-based taxonomy for IoT sensors. Advances in Transdisciplinary Engineering, 4, 249–258. https://doi.org/10.3233/978-1-61499-703-0-249

[2] CCD vs. CMOS, https://www.teledynedalsa.com/en/learn/knowledge-center/ccd-vs-cmos/

[3] Valuing the use of spectrum in the EU. An independent assessment for the GSMA, https://www.gsma.com/spectrum/wp-content/uploads/2013/06/Economic-Value-of-Spectrum-Use-in-Europe_Junev4.1.pdf

[4] Md Habibul Islam et al., Spectrum Survey in Singapore: Occupancy Measurements and Analyses, in Proceedings of CrownCom'08, 2008.

[5] S. D'Iitri and M. McHenry, Dynamic spectrum access moves to the forefront, in Defence Electronics April 2008

[6] V. Valenta, R. Maršálek, G. Baudoin, M. Villegas, M. Suarez and F. Robert, "Survey on spectrum utilization in Europe: Measurements, analyses and observations," 2010 Proceedings of the Fifth International Conference on Cognitive Radio Oriented Wireless Networks and Communications, Cannes, 2010

[7] I. Akyildiz, W. Lee, M. Vuran and S. Mohanty, “NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey,” Computer Networks, Vol. 50, No. 13, pp. 2127-2159, 2006

[8] Efficient spectrum utilization: A cognitive approach. Available from: https://www.researchgate.net/publication/229034409_Efficient_spectrum_utilization_A_cognitive_approach [accessed Nov 01 2019].

[9] Shao-qian, WANG Jun LI. "Cognitive Radio: Principle, Technology and Tendency [J]." ZTE Communications 3 (2007).

[10] P. Sofotasios, E. Rebeiz, L. Zhang, T. Tsiftsis, D.Cabric, and S. Freear, “Energy detection based spectrumsensing overκ-μandκ-μextreme fading channels,”IEEE Trans. Veh. Technol., vol. 62, no. 3, pp. 1031–1040, Mar. 2013.

[11] X. Zhang, F. Gao, R. Chai, and T Jiang, “Matched filterbased spectrum sensing when primary user has multiple power levels,” China Commun., vol. 12, no. 2, pp. 21–31, Feb. 2015.

[12] M. Yang, Y. Li, X. Liu, and W. Tang, “Cyclostationaryfeature detection based spectrum sensing algorithm under complicated electromagnetic environment in cognitive radio networks,” China Commun., vol. 12, no.9, pp. 35–44, Sep. 2015.

[13] M. Jin, Q. Guo, J. Xi, Y. Li, Y. Yu, and D. Huang,“Spectrum sensing using weighted covariance matrix inRayleigh fading channels,” IEEE Trans. Veh. Technol.,vol. 64, no. 11, pp. 5137–5148, Nov. 2015.

[14] S. Jindal, D. Dass, and R. Gangopadhyay, “Waveletbased spectrum sensing in a multipath Rayleigh fadingchannel,” in Proc. IEEE Twentieth Nat. Conf. Commun.(NCC), 2014, pp. 1–6

[15] A. Margoosian, J. Abouei, and K. N. Plataniotis, “Accurate kernel-based spectrum sensingfor Gaussian and non-Gaussian noise models,” inProc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP), Apr. 19–24, 2015, pp. 3152–3156

[16] IEEE 802.22 Working Group on Wireless Regional Area Networks, Enabling Broadband Wireless Access Using Cognitive Radio Technology and Spectrum Sharing in White Spaces, http://www.ieee802.org/22/

[17] Jia, J.; He, Z.; Kuang, J.; Wang, H. Analysis of key technologies for cognitive radio based wireless sensor networks. In Proceedings of the 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM), Chengdu, China, September 2010

[18] Gislason, D. (2008). Zigbee wireless networking. 1st ed. Amsterdam [etc.]: Elsevier, p.303.

[19] https://www.digicatapult.org.uk/

[20] https://www.thethingsnetwork.org/

[21] https://www.digicatapult.org.uk/projects/things-connected/

[22] https://www.nbc-2.com/story/39751041/z-wave-alliance-hosts-interactive-smart-home-pavilion-at-ces-2019

[23] Report ITU-R SM.2152,m "Definitions of Software Defined Radio (SDR) and Cognitive Radio System (CRS), 09/2009, https://www.itu.int/pub/R-REP-SM.2152-2009

[24] Akeela, Rami, and Behnam Dezfouli. "Software-defined Radios: Architecture, state-of-the-art, and challenges." Computer Communications 128 (2018): 106-125.

[25] J. J. Divya lakshmi, Rangaiah. L, "Cognitive Radio Principles and Spectrum Sensing", International Journal of Engineering and Advanced Technology (IJEAT), ISSN: 2249 – 8958, Volume-8 Issue-6, August 2019

[26] Goldsmith, A. J., Jafar, S. A., Maric, I., & Srinivasa, S. (2009). Breaking spectrum gridlock with cognitive radios: An information theoretic perspective. Proceedings of the IEEE, 97(5), 894-914.

[27] Pérez-Romero, J., Raschellà, A., Sallent, O., & Umbert, A. (2015). A belief-based decision-making framework for spectrum selection in cognitive radio networks. IEEE Transactions on Vehicular Technology, 65(10), 8283-8296.

[28] Raschellà, A., Pérez-Romero, J., Sallent, O., & Umbert, A. (2013, July). On the use of POMDP for spectrum selection in cognitive radio networks. In 8th IEEE International Conference on Cognitive Radio Oriented Wireless Networks (pp. 19-24)

[29] Leaves, P. A. U. L., Ghaheri-Niri, S., Tafazolli, R., Christodoulides, L., Sammut, T., Staht, W., & Huschke, J. (2001). Dynamic spectrum allocation in a multi-radio environment: Concept and algorithm.

[30] Khairullah, E. F., & Chatterjee, M. (2019). PreDA: Preference-based double auction for spectrum allocation in heterogeneous DSA networks. Computer Communications, 133, 41-50.

[31] He, A., Bae, K. K., Newman, T. R., Gaeddert, J., Kim, K., Menon, R.,... & Tranter, W. H. (2010). A survey of artificial intelligence for cognitive radios. IEEE Transactions on Vehicular Technology, 59(4), 1578-1592.

[32] Huang, Q., Xie, X., Tang, H., Hong, T., Kadoch, M., Nguyen, K. K., & Cheriet, M. (2019). Machine-Learning-Based Cognitive Spectrum Assignment for 5G URLLC Applications. IEEE Network, 33(4), 30-35.

[33] Agarwal, A., Dubey, S., Khan, M. A., Gangopadhyay, R., & Debnath, S. (2016, June). Learning based primary user activity prediction in cognitive radio networks for efficient dynamic spectrum access. In 2016 International Conference on Signal Processing and Communications (SPCOM) (pp. 1-5). IEEE.

[34] Agarwal, A., Gangopadhyay, R., Dubey, S., Debnath, S., & Khan, M. A. (2018). Learning-based predictive dynamic spectrum access framework: a practical perspective for enhanced QoE of secondary users. IET Communications, 12(18), 2243-2252.

[35] Agarwal, A., Sengar, A. S., & Gangopadhyay, R. (2018). Spectrum Occupancy Prediction for Realistic Traffic Scenarios: Time Series versus Learning-Based Models. Journal of Communications and Information Networks, 3(2), 44-51.

[36] A. Custura, R. Secchi et al. (2018) "Exploring DSCP modification pathologies in the internet" Computer Communications, 127, 9 2018, Elsevier

[37] https://www.itu.int/rec/T-REC-P.910/en

[38] https://www.itu.int/rec/T-REC-P.800-199608-I

[39] https://www.itu.int/rec/T-REC-P.863

[40] https://www.sdxcentral.com/networking/sdn/

[41] L. Yang (Intel Corp.), R. Dantu (Univ. of North Texas), T. Anderson (Intel Corp.) & R. Gopal (Nokia.) (April 2004). "Forwarding and Control Element Separation (ForCES) Framework"

[42] Martìn Casado, Michael J. Freedman, Justin Pettit, Jianying Luo, and Nick McKeown (Stanford University) (August 2007). "Ethane: Taking Control of the Enterprise"

[43] McKeown, Nick, et al. "OpenFlow: enabling innovation in campus networks." ACM SIGCOMM Computer Communication Review 38.2 (2008): 69-74.

[44] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker. (July 2008). "NOX: Towards an Operating System for Networks"

[45] Enns, Rob; Björklund, Martin; Schönwälder, Jürgen; Bierman, Andy (2011). Network Configuration Protocol (NETCONF)

[46] A. Rodriguez-Natal et al., "LISP: A southbound SDN protocol?", IEEE Commun. Mag., vol. 53, no. 7, pp. 201-207, Jul. 2015

[47] T. Koponen et al, "Onix: A Distributed Control Platform for Large scale Production Networks," proceedings USENIX, ser. OSDI'10, Vancouver, Canada, 2010

[48] N. Saritakumar Adarsh V Srinivasan Elfreda Albert S. Subha Rani, "Performance Evaluation of Pox Controller for Software Defined Networks" July 2019

[49] David Erickson (Stanford University), "The Beacon OpenFlow Controller"

[50] https://www.opendaylight.org/

[51] https://searchnetworking.techtarget.com/tip/REST-APIs-in-SDN-An-introduction-for-network-engineers

[52] S. K. Tayyaba, M. A. Shah, O. A. Khan, and A. W. Ahmed, "Softwaredefined network (sdn) based internet of things (iot): A road ahead,"

[53] Gudipati, A., Perry, D., Li, L. and Katti, S. 2013. SoftRAN: Software defined radio access network. Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking (2013).

[54] Akyildiz, I., Wang, P. and Lin, S. 2015. SoftAir: A software defined networking architecture for 5G wireless systems. Computer Networks. 85, (2015), 1-18

[55] Galluccio, L., Milardo, S., Morabito, G. and Palazzo, S. 2015. SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for WIreless SEnsor networks. IEEE Conference on Computer Communications (INFOCOM) (2015), 513-521

[56] Miyazaki, T., Yamaguch, S., Kobayashi, K., Kitamichi, J., Guo, S., Tsukahara, T. and Hayashi, T. 2014. A software defined wireless sensor network. International Conference on Computing, Networking and Communications (ICNC) (2014), 847-852

[57] https://www.sdxcentral.com/networking/virtualization/definitions/what-is-network-service-chaining/

[58] S. Bhelekar, M. Iyer, G. Mehta and S. Chaudhari, "Dynamic load balancing strategy in software-defined networking," 2017 International Conference on Trends in Electronics and Informatics (ICEI), Tirunelveli, 2017, pp. 875-878.

[59] Gkatzikis, Lazaros & Paris, Stefano & Stiakogiannakis, Ioannis & Chouvardas, Symeon. (2016). Bandwidth Calendaring: Dynamic Services Scheduling over Software Defined Networks. 10.1109/ICC.2016.7510888.

[60] https://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.02.01_60/gs_NFV002v010201p.pdf

[61] F. Bari, S. R. Chowdhury, R. Ahmed, and R. Boutaba, “On Orchestrating Virtual Network Functions,” in 11th International Conference on Network and Service Management (CNSM), no. Section III. IEEE, 2015, pp. 50— -56.

[62] B. Addis, D. Belabed, M. Bouet, and S. Secci, “Virtual Network Functions Placement and Routing Optimization,” in IEEE 4th International Conference on Cloud Networking (CloudNet), 2015, pp. 171–177.

[63] A. Baumgartner, V. S. Reddy, and T. Bauschert, “Mobile core network virtualization: A model for combined virtual core network function placement and topology optimization,” in 1st IEEE Conference on Network Softwarization (NETSOFT), 2015

[64] T. Taleb, M. Bagaa, and A. Ksentini, “User mobility-aware Virtual Network Function placement for Virtual 5G Network Infrastructure,” in IEEE International Conference on Communications (ICC), 2015

[65] R. Riggio, A. Bradai, T. Rasheed, J. Schulz-Zander, S. Kuklinski, and T. Ahmed, “Virtual Network Functions Orchestration in Wireless Networks,” in 11th IEEE International Conference on Network and Service Management (CNSM), 2015

[66] https://searchnetworking.techtarget.com/definition/data-plane-DP

[67] https://www.semanticscholar.org/paper/Software-Defined-Network-(SDN)-Data-Plane-Security%3A-Shaghaghi-K%C3%A2afar/d202e6f5438043839f36ee9c5b3de3738cfa826c/figure/0

[68] Jawhar I et.al., Networking architectures and protocols for smart city systems, 2018, Jawhar et al. Journal of Internet Services and Applications (2018) 9:26 https://doi.org/10.1186/s13174-018-0097-0

[69] Zanella A, Bui N, Castellani A, Vangelista L, Zorzi M (2014) Internet of things for smart cities. IEEE Internet Things J 1(1):22–32

[70] Jin J, Gubbi J, Luo T, Palaniswami M (2012) Network architecture and QoS issues in the internet of things for a smart city, IEEE International Symposium on Communications and Information Technologies (ISCIT), 2012

[71] https://www.sdxcentral.com/networking/virtualization/definitions/what-is-network-service-chaining/

[72] https://www.google.com/search?q=smart+grid&source=lnms&tbm=isch&sa=X&ved=2ahUKEwj9y-K325bmAhXCsKQKHfBuDa8Q_AUoAXoECBIQAw&biw=1536&bih=722#imgrc=unyj1xQZobCtAM:

[73] https://www.ciena.com/insights/what-is/What-Is-an-Autonomous-Network.html

[74] https://microcontrollerslab.com/difference-between-actuator-and-sensor/

[75] http://www.automatedbuildings.com/news/jan06/reviews/ciscowhitepaper.htm

[76] https://whatis.techtarget.com/definition/ubiquitous-networking

[77] https://whatis.techtarget.com/definition/service-chaining

[78] https://www.researchgate.net/publication/303229562_Bandwidth_Calendaring_Dynamic_Services_Scheduling_over_Software_Defined_Networks

[79] https://en.wikipedia.org/wiki/Load_management

[80] Gkatzikis L., et al., Bandwidth Calendaring: Dynamic Services, Scheduling over Software Defined Networks, 2016, https://www.researchgate.net/publication/303229562

[81] Deepika M. S., Rama Mohan babu, An Approach to Effective Bandwidth Utilization using Software Define Networking, 2014, ISSN: 0975-9646, https://pdfs.semanticscholar.org/0c8a/9decc75f522f388167b9605a8feb96eba169.pdf

[82] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ImageNet classification with deep convolutional neural networks, Advances in Neural Information Processing Systems (NIPS) 25, 2012

[83] S. Lowry et al., Visual Place Recognition: A Survey, IEEE Transactions on Robotics, vol. 32, no. 1, pp. 1-19, Feb 2016

[84] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla and J. Sivic, NetVLAD: CNN Architecture for Weakly Supervised Place Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 6, pp. 1437-1451, 1 June 2018

[85] Localization demo for central Cambridge, UK http://mi.eng.cam.ac.uk/projects/relocalisation/

[86] Y. Tian, C. Chen and M. Shah, Cross-View Image Matching for Geo-Localization in Urban Environments, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, 2017

[87] X. Yu et al., VLASE: Vehicle Localization by Aggregating Semantic Edges, 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, 2018

[88] VLASE aerial video, YouTube, https://www.youtube.com/watch?v=IKZXZmmdtiA

[89] Paulo Costa et al., Landmarks detection to assist the navigation of visually impaired people, International Conference on Human-Computer Interaction (HCI), 2011

[90] Yitao Hu et al, ALPS: Accurate Landmark Positioning at City Scales, ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp), July 2016

[91] Google-Landmarks: A New Dataset and Challenge for Landmark Recognition, Google AI Blog, March 2018, https://ai.googleblog.com/2018/03/google-landmarks-new-dataset-and.html

[92] Announcing Google-Landmarks-v2: An Improved Dataset for Landmark Recognition & Retrieval, Google AI Blog, May 2019, https://ai.googleblog.com/2019/05/announcing-google-landmarks-v2-improved.html