Università di Pisa and Scuola Superiore Sant'Anna

Depts. of Computer Science and Information Engineering and
Scuola Superiore Sant'Anna
Master Degree in Computer Science and Networking

Master Thesis

Design and implementation of an ITS station to bridge automotive and IoT systems

Candidate
Michele Carignani

Supervisors
Prof. Piero Castoldi
Dr. Paolo Pagano

Examiner
Dr. Nicola Tonellotto

Academic Year 2014/2015

To my grandparents,for the invaluable
heritage of love they left to me.

To my parents, for their constant
trust, support, and inspiration.

To my sisters and brother,
who taught me the beauty of sharing.

Contents

Chapter 1

Introduction

1.1 The Internet of Things

Generally speaking the terms Internet of Things (IoT) refer to the implementation on constrained-embedded-mobile devices of Internet technologies, notably the standard communication protocols and most of all the Internet Protocol.

More appropriately, the Internet of Things is defined by the International Telecommunication Union as

> "a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies"[1]

The Internet of Things is being recognized as one of the pillars of the Future Internet, along with the Internet of Content and Knowledge, the Internet by and for People and the Internet of Services. It is more and more entering many sectors from industrial control and quality assurance, to smart cities, health and personal wellness. Its building block are smart devices, i.e. a variety of electronic tools featuring some computing technologies (from FPGAs to micro-controllers to SOMs with many-cores architectures) with enough resources capable to embed communication protocols.

The realization of the IoT evolution and remaining challenges involve the development of standards and protocols as well as their promotion and imple-

[1] see `http://www.itu.int/en/ITU-T/gsi/iot/Pages/default.aspx`

mentation on behalf of the industry. Regarding communication and connectivity stacks a strong contribution comes from Open Sources Initiatives that produced well affirmed protocols (such as CoAP, MQTT, 6LoWPAN being standardized by IETF [1].)

1.2 Intelligent Transport Systems

The world of automotive technologies is moving fast towards a pervasive digitalization. All vehicles (cars, trucks, trains, planes) already feature on-board hundreds of Micro-Controller Units (MCUs) that support vehicle mobility functions. This equipment is nowadays mostly specific to the environment it is applied and most of it is based on proprietary technologies since it is developed by the manufactures of the vehicles themselves.

A big revolution is approaching: a new set of services is going to be available on vehicles, to support infotainment, power efficiency, remote monitoring and control, automated driving, cooperative safety. These services are recognized all over the World under the label Intelligent Transport Systems (ITS), since the integration of these services enables the machines to extract knowledge and support more powerful decision-making processes.

This revolution is calling for standards to be set on which vendors and manufactures can build their products and solutions.

Moreover the vehicles in the future will communicate and cooperate not only with other vehicles but also with remote control centers (e.g. in smart cities) to monitor traffic and enhance efficiency.

Big issues in security, privacy and resource waste are arising with respect to these new systems and the role of international standardizing bodies is to agree on technologies and processes while preserving the vitality of the market.

A recent set of standards for intelligent transport systems has been released by the European Telecommunication Standard Institute (ETSI) in April 2014 and is being fine-tuned and tested on real testbeds through the so called “ETSI Plugtests” campaign all over Europe.

Many Framework 7 projects were carried on by universities and companies in Europe for Research and Development in the ITS sector.

With the new standards being set, a new call arises: common and open tools (e.g. components, devices, software) are required to speedup the development and to enlarge the range of applications in the automotive sector.

1.3 The opportunity for a reciprocal inclusion

The ITS are a new field where many actors are involved (from vehicle vendors to governmental and administrative bodies) fostering applications for safety, efficiency and smart cities.

The standardized ITS are designed to be compliant and to embed many telecommunication technologies but, at the moment of writing this thesis, a state of art solution to include in ITS the technologies and protocols that are widely spread and used in the IoT world is missing.

On the other hand, many application would benefit from the usage of on board wireless sensors, connecting for example with external wireless devices and so on. Therefore the authors argue that the ITS, especially the vehicle on board equipment, should be able to cooperate with the embedded and constrained devices of the IoT.

1.4 A novel integrated approach

In the next chapters we propose the integration between these two worlds by designing a new system. The developed mockup is tested against a loading zone management service in urban environment.

The novel contribution of this thesis is summarized as follows:

a. Following the guidelines of the standards in ITS - i.e. as a ETSI ITS-S a new software architecture has been designed and developed. It is capable to support the latest standardized communication media and protocols, with a minimal footprint on computational resources (in order to be deployed easily on constrained and "embedded" devices) exploiting them with the maximum efficiency.

b. The proposed hardware-software architecture has been tested on-field, since it has been used to implement use cases of parking slot management for the

URBELOG project[2], funded by the Italian Ministry of Education, Universities and Research and coordinated by Telecom Italia. The project aims at building and validating a virtuous freight transport system to enable a reasonable, economic, efficient and sustainable last mile logistics service.

1.5 Last mile logistics and the loading zone application

The use cases implemented are meaningful in the context of both the Internet of Things, and its usage in the Smart City domain, and of the Intelligent Transport Systems. Indeed, a central part in the standardization process brought about by ETSI is the definition of possible use cases and application for Intelligent Transport Systems. The results of this activity are included in the so called Basic Set of Applications (BSA) [2].

The document defines the application requirements to be included in the BSA and then lists a rich catalogue of already selected applications. The applications in the catalogue are divided into three main categories: road safety, traffic efficiency and other applications.

Within the other applications category, an application called "Loading zone management" is described . This application belongs to the "Community services" application category and to the more general application class of "global Internet services". Applications in the global Internet services classes advertise and provide on-demand information to vehicles passing by on either a commercial or a non-commercial basis. These services may include infotainment, comfort and vehicle or life service management. Co-operative local services are provided from within the ITS network infrastructure. Global Internet services are acquired from providers in the wider Internet.

The goal with this Use Case is to support the driver, fleet manager and road operator (including parking zone operator) in the booking, monitoring and management of the urban parking lots appointed to freight services. These activities can be those of loading/unloading of both heavy vehicles and (for parcel operators) smaller vehicles.

[2] see `http://www.urbelog.it`

The communication modes considered by the standard are:

- Vehicular Ad-hoc NETwork (VANET) based on IEEE 802.11p (which will be presented in section 3.4) and

- Access to Internet for Machine to Machine communication.

For the “Loading zone management” application the main requirement recognized by the standards is the capability for a road side device to:

a. broadcast I2V Co-operative Awareness Messages (see section 3.8.3) informing on its capabilities to liaise with parking booking centre,

b. the capability for this RSU to establish a P2P unicast session with any requesting vehicle to fully process the parking booking service,

c. the capability for concerned vehicles to receive I2V Co-operative Awareness messages process them and establish any requested P2P unicast sessions with the local road side unit to fully process the parking booking request,

d. foresee a I2V minimum CAMs frequency of 1 Hz and a maximum latency time is 500 ms as from the ETSI specifications for this kind of service.

1.6 The URBELOG project

One of the goal of the URBELOG project is contributing to better integrate ITS systems with the Internet of Things in the field of Smart Cities.

URBELOG focuses the attention on last mile logistics in smart cities: the management of goods delivery within the green areas in a city can and should be optimized in order to minimize environmental indicators relevant for pollution like the amount of harmful emissions. The mission of URBELOG is thus to coordinate and to enforce best practices in logistics through the collaboration among all playing actors (city managers and administration, private courier companies, communication operators).

Within the project, a well tuned system of ”ecological credits” is envisioned in order to provide access to the delivery vehicle into the limited traffic zone. Credits are decreased on a emission per kilogram delivered basis. The technological challenge here comes: the need for a ICT infrastructure capable of collecting real time data about vehicles, their position, their emissions and the transported cargo.

Moreover, in order to optimize traffic routing and delivery time efficiency, a system of smart loading zone parking areas is envisioned for the private companies to be used. According to the project these areas are reserved to the couriers taking part in the partnership and a control center which should be able to follow the current status to support routing decisions. Therefore a monitoring system able to collect data about the occupancy status of the reserved parking slots is needed.

We emphasize that this requirement naturally fits with the application expected by ETSI about the service “Loading zone management”. In this vision, the URBELOG project can be really seen as an implementation effort to match the standards. Thus starting from what is already standardized it will be useful to exploit the next results from URBELOG project to improve the ETSI ITS-G5 standards at least as regards the “Loading zone management” use case.

In conclusion, the URBELOG project requires on board and road side equipment to collect and manage real time data. It is considered a natural, standard compliant application for a congruent and robust integration of solutions from the field of the Internet of Things and those from Intelligent Transport Systems. The concepts and tools developed in this thesis enabled the actual development of such devices in a structured way. These devices, at the moment of writing, are already set up and tested in a third party lab showing the effectiveness of this implementation.

Chapter 2

IoT technologies

2.1 Definitions and application domains

The concept of *smart object* is the fundamental building block in the IoT vision. Smart objects are not only supposed to collect information and to interact with or to control physical objects but also to exchange machine readable information.

This new kind of interaction among objects is featuring cyber-physical systems (CPS), a next generation of embedded ICT systems where computation and networking are fully integrated with physical processes. Smart objects are at the heart of this processes, control and manage providing them with more efficiency, reliability, flexibility and security.

Smart environments are therefore designed together with the physical components to maximize the overall efficiency, in contrast with classic embedded systems where the goal is to include electronics, computing, and communication devices separated from the physical world.

In May 2013, The McKinsey Global Institute has presented its vision on disruptive technologies in Cyber-Physical-Systems [3]: they are identified in automation of knowledge work, the Internet of Things, advanced robotics, and autonomous/ near-autonomous vehicles. Among them, the Internet of Things (IoT), with an estimated value of 36 trillion of dollars, is considered the CPS paradigm with the highest economic impact.

The IoT refers to an emerging paradigm consisting of a continuum of uniquely addressable things communicating one another to form a worldwide dynamic network. The origin of IoT has been attributed to members of the Auto-ID Center at MIT, the development community of the Radio-Frequency Identification (RFID), around 2000. The term was coined by Kevin Ashton, one of the founders of the original Auto-ID Center at MIT, who introduced it in 1999 during a presentation held at Procter & Gamble (P&G).

A thing can be any real/physical object (e.g., RFID, sensor, actuator, smart item) but also a virtual digital entity, which moves in time and space and can be uniquely identified by assigned identification numbers, names and/or location addresses.

The objects are no more dedicated to a unique function.

It is envisaged that the number of connected things will exceed 7 trillion by 2025, with an estimate of about 1000 devices per person.

The city is the economic and social life core of a country. Today, half of the global population is concentrated in the cities and consume its resources (e.g., light, water) every day.

Efficiency at multiple levels aims to achieve:

- a more aware and optimized usage of the offered resources,
- a minimization of environmental impact, for example by reducing CO2 emissions, and
- a tangible increase in the life quality in terms of safety, health, and wellness.

The applications envisioned to be powered by the IoT are many and they range from hardware, architecture, communication, discovery, data processing, data and network management, power and energy storage, security and privacy to cite a few of them.

ITU explains its vision: "a new dimension has been added to the world of information and communication technologies (ICTs): from anytime, any place connectivity for anyone, we will now have connectivity for anything. Connections will multiply and create an entirely new dynamic network of networks - an Internet of Things". Therefore not only RFID but any object can constitute the underlying fabric of the Internet of Things.

Three categories of visions for the IoT are:

- Things oriented: focus on the object, the paradigm leads to interconnect them
- Internet Oriented: using IP to easily connect anything, adapt IP so it can be used on constrained devices
- Semantic oriented: focus on semantic technologies to interconnect, manage and store the information coming from the real world.

A different definition comes from the Cluster of European Research Projects for IoT (CERP-IoT): a vision that blends the concepts of Ubiquitous Computing, Pervasive Computing and Ambient Intelligence and enhances them. They focus the attention on the self-capacities the network has based on the objects by associating virtual profiles and identities. Also they focus on seamless network transparency. Therefore not only “Anytime” “Anywhere” with “Anyone” and “Anything”, but also use any type of location or network and any available service. Hence, two additional concepts, i.e., “Any path/network” and “Any service”, are introduced to complete the picture forming the so-called *6A vision*. Later, two other concepts were also included: the massive user-interaction of the web 2.0 and the need for self-sustainability of the systems.

2.1.1 Machine to Machine paradigm

Machine to Machine (M2M) communications are also taking part in this scenario: the conservative view is to consider M2M a natural evolution of the embedded systems networks; the more visionary outlook sees M2M as a possible new revolution, similar to the birth of information technology itself.

The following definition was provided by the ETSI Technical Committee on Machine-to-Machine Communications (ETSI TC M2M): “Machine-to-Machine (M2M) communication is the communication between two or more entities that do not necessarily need any direct human intervention”.

2.1.2 Driver technologies

Nowadays a ”silo” approach is mostly adopted to support Cyber-Physical-Systems, i.e. a vertical proprietary ICT infrastructure providing dedicated services in a manner that is incompatible or not integrated with other systems.

instead, the evolution of the Internet especially for the Internet of Things infrastructure is going towards the accomplishment of common operational platforms that will manage the network and the services, and will enable new application abstracting data spread across a wide range of sources.

The architecture of this novel infrastructure can be composed of threes functional layers:

- collection layer: sensing the environment, collecting physical real time data, reconstructing a perception of it;
- transmission layers;
- process, management and utilization layer.

In the data collection layer, actual data acquisition is encompassed by using different sensing technologies attached to sensors, cameras, GPS terminals, while data collection is generally accomplished by short range communications, which could be open source standard solutions (e.g IEEE 802.15.4, 6LowPAN, Wireless M-BUS, Dash7), as well as proprietary solutions (e.g. Bluetooth, ZigBee, Z-Wave, ANT).

List of technologies for the collection layer

- RFID
- Wireless Sensor Networks (802.15.4)
- Near Field Communication
- Blue tooth and Blue tooth low energy

Transmission layer technologies

- Ethernet (IEEE 802.3)
- WLAN (IEEE 802.11a/g/n)
- Broadband technologies (xDSL)
- Cellular Networks (2/3/4G and in the near future 5G)

- Satellite communications

Wireless technologies, due to their flexibility, will be the main communication paradigm for the IoT.

Process management and utilization

In this layer, information flows are processed and then forwarded to applications. The Service Platform & Enabler covers a fundamental role for managing the above operations. It is crucial for hiding the heterogeneity of hardware, software, data formats, technologies and communication protocols characterizing IoT. It is responsible for abstracting all the features of objects, network, and services, and for offering a loose coupling of components.

Additional features are service discovery and service composition.

To address the above challenges, the Service-Oriented Architecture (SOA) concept can be inherited and applied to IoT but also Cloud computing could improve scalability and robustness of the process management and utilization layer.

2.1.3 Key features

Among the general features and requirements of such a complex and dynamic system as the IoT, heterogeneity and scalability will be of primary importance. Solutions to cope with the above requirements must be sought at architectural level, at naming/ identification/addressing level, at communication level, and at level of object name/code mapping services. Effectiveness and cost minimization is another requirement both in operational cost, energy efficiency and production costs.

An important requirement of IoT powered smart objects are self-capabilities which means low o none human intervention needed for operations. Self-capabilities are enabled with high degree of configuration autonomy, self- organization and self-adaptation to various scenarios, self- reaction to events and stimuli to which objects are subjected, and self-processing of the huge amounts of exchanged data.

For real time applications, a general requirement will be the observance of Quality of Service.

The last concern is about security: the safe and reliable operation of IoT connected devices should occur in secure environment for both the infrastructure and its users in all its forms (authentication, integrity, confidentiality, trustworthiness).

2.2 Application domains

The Internet of Things paradigm and technologies can have applications in nearly every field because of situated sensing and tailored services. We can identify three major application domains: the industrial domain, the smart city domain, and the health and well-being domain. These three domains actually overlaps since some applications can serve more than one of them.

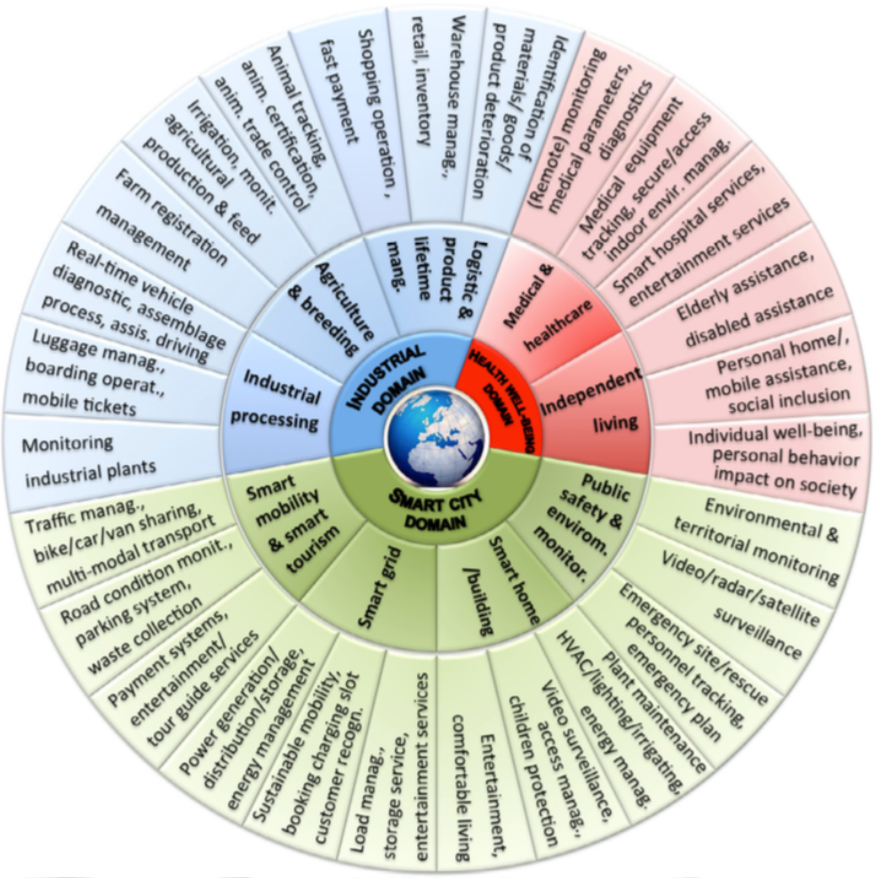

Figure 2.1: A categorized list of applications for the Internet of Thing. Source *Borgia et al.* [4]

2.2.1 Industrial domain

IoT can be applied in almost every industrial process among which logistics, manufacturing, monitoring of production, service sector, banking, financial governmental authorities, intermediaries, etc.

For example, regarding logistics and product lifetime management, given the huge amount of products manufactured and created in the world, an important need of quality control and assurance is more and more arising. An answer to that need is provided by one of the earliest exploitation of the Internet of Things (as it was in the original vision of) namely attaching RFID to objects for identification purposes or to track their life time. This means monitoring the production process among the *entire multi-domain multi-parties* supply chain.

A second, more complex approach is the pervasive usage of smart sensors to monitor the status of goods in real time. This enables the vision of the *smart plant* where every point of the production line is monitored at every time in a scalable and sustainable way, which means by getting sensed data only when it is interesting with a low-cost system.

This vision is applicable also to the segment of agriculture and breeding, through identification and monitoring of animals, plants, fruits. Monitoring and controlling agricultural production and feed (e.g., presence of OMGs, additives, melanin) by using advanced sensor systems is for sure the challenge in agriculture in the next years.

Other emerging applications of IoT in the industrial field are real-time vehicle diagnostic and monitoring industrial plants, for instance to reduce the number of accidents, especially in case of high-risk plants.

2.2.2 Smart city domain

Environmental sustainability of our cities and the people's quality of life. Emphasis is on energy and how to manage it efficiently, and on seeking smart solutions to enjoy the personal stay.

2.2.3 Health - well being domain

Body Area Networks (BANs), formed by wearable devices connected to each other, allow doctors to continue the remote patient's monitoring out of the hospital. Accurate tracking and management of medical instrumentations and materials can be achieved through the usage of smart labels. Similar applications will be exploited by visual impaired persons to increase their ability to move in the city. Positive feedbacks, for example notifying the number of kilometers (calorie) covered (burnt) during a walk, and the positive impact on their health, will motivate people to repeat daily the same activity.

2.3 Communication standards

We want now to recall the major leading standards used to interconnect the smart objects at the collection layers. Within the other layers the most common standards (mainly TCP/IP stack) are used but when dealing with constrained, embedded and often battery powered devices the communication standards developed for the Wireless Sensor Networks (WSN) are used. We will just list the main features of this protocols, since they will be used in our architecture to design the IoT enabled ITS station.

We recall that, according to the ETSI definition, a *Device* is a piece of equipment that may collect a set of actuators and sensors with embedded electronic computing and communication capability, while a *Gateway* aims at translating and transferring information between two or more communicating entities, or at performing some routing and multiplexing function between the communicating entities

A more detailed presentation of the protocols in the IoT stack can be found in *Pacini* [5].

2.3.1 IEEE 802.15.4 physical and link layer protocol

The IEEE 802.15.4 standard developed by the IEEE 802.15.4 Task Group within the IEEE specifies the physical layer and media access control layer for Low-Rate Wireless Personal Area Networks (LR-WPAN) providing node-to-node frame delivery between devices within reachable distance from each other. The first version was completed in May 2003.

In contrast to Wireless Local Area Network (WLAN), which is standardized within IEEE 802.11 family, LR-WPANs focus on short-range operation, low-data-rate, energy-efficiency, and low-cost. It must be stressed that the main design goal has been energy efficiency, whereas real-time aspects were not a primary concern although soft real time can be reached by using the beacon enable communication mode provided by the standard.

In IEEE 802.15.4 communications, the 2.45 Ghz ISM band, where 16 channels spaced by 5MHz are defined, is used so that it is worldwide permitted to perform R&D using a public license-free communication channel. The protocol permits transmission of frames of length up to 128 bytes.

Addresses can be of two types: either globally unique extended IEEE EUI-64 bit addresses, or short 16 bit addresses which refer to a given PAN network. The IEEE 802.15.4 MAC layer is designed for a network with a star topology.

2.3.2 6LoWPAN networking protocol

IETF 6LoWPAN2 standard aims at integrating existing IP based infrastructures and sensor networks by specifying compression policies for sending IPv6 in IEEE 802.15.4 based networks. The 6LoWPAN concept comes from the idea that the Internet Protocol could and should be applied even to the smallest device, and that low-power devices with limited processing capabilities should be able to participate in the envisioned Internet of Things. 6LoWPAN defines the frame format for the transmission of IPv6 packets as well as the formation of IPv6 link-local addresses on top of IEEE 802.15.4 networks.

Since IPv6 requires support of packet sizes much larger than the largest IEEE 802.15.4 frame size, respectively of 1280 and 128 bytes, and since the IPv6 40 bytes length header would introduce a high overhead, an adaptation layer is defined. This standard also defines aggressive mechanisms for header compression required to make IPv6 practical on IEEE 802.15.4 networks, and the provisions required for packet delivery in IEEE 802.15.4 mesh networks. Regarding available public documents, RFC 49193 describes an overview, assumptions, problem statement, and goals, while RFC 49444 (now RFC 62825) defines the standard itself.

2.3.3 Routing Protocol for Low Power and Lossy Networks

Routing issues are very challenging for 6LoWPAN, given the low-power and lossy radio-links, the battery supplied nodes, the multi-hop mesh topologies, and the frequent topology changes due to mobility. Successful solutions should take into account the specific application requirements, along with IPv6 behavior and 6LoWPAN mechanisms.

Routing Protocol for Low-power and Lossy networks RPL, defined in RFC 65506, can be considered to be state-of-the-art routing algorithm developed by the networking community. RPL has been proposed by the IETF Routing over Low-power and Lossy networks Working Group (ROLL) as a standard routing protocol for 6LoWPAN, since existing routing protocols do not satisfy all the requirements for Low power and Lossy Networks (LLNs). RPL organizes a topology as a Directed Acyclic Graph (DAG) that is partitioned into one or more Destination Oriented DAGs (DODAGs), one DODAG per sink. It forms a non-transitive, non-broadcast multiple-access (NBMA), flexible network topologies upon which it computes routes.

2.3.4 Constrained Application Protocol (CoAP)

CoAP is the main standard utilized at the application layer within IoT. CoAP has being standardized within the CoRE working group of the IETF, which aims at providing a REST-based framework for resource-oriented applications optimized for constrained IP networks and devices, by designing a protocol stack able to cope with limited packet sizes, low-energy devices and unreliable channels.

CoAP is based on a REST- ful architectural approach, and shares the same objectives and the intrinsic limitation listed above. It is designed for easy stateless mapping with HTTP and provides easy M2M interaction. HTTP compatibility is obtained by maintaining the same interaction model, but using a subset of the HTTP methods.

CoAP features some capability that are not supported by HTTP, tough, but are extremely useful when dealing with lossy communication and power critical applications such as built-in discovery, multicast support and asynchronous message exchange (it implements push communication via a notification mechanism).

Nodes supporting CoAP provide flexible services over any IP network upon UDP (the User Datagram Protocol), and they also provide a solid communication framework to connect sensor nodes to the Internet. Any HTTP client or server can interoperate with CoAP ready endpoints by simply installing a translation proxy between the two devices. This will not be a burden for the proxy, since these translation operations have been designed not to be time and computationally demanding. Also, CoAP features a transaction layer between the application protocol and UDP to provide basic reliability and session matching support.

Chapter 3

C-ITS and ITS Station architecture

As we have seen automotive systems are a relevant domain within the Internet of Things, which in turn is one of the pillars of the Future Internet.We will present in this chapter a standardized architecture for Intelligent Transport Systems.

We will first revise current automotive technologies already on field and later in the chapter we will try to collect the main features of how this systems should be designed. These concept have been synthesized and became normative in Europe mostly by the efforts of ETSI, summing the experience and results from other international standardization bodies, manufacturer and vendor alliances.

3.1 Automotive technologies

There is a plentitude of IoT technologies and standards that could be and are used for mobility applications and it is anticipated that new technologies will appear permanently. For sure, the wireless connections among vehicles and between vehicles and infrastructure will play a pivotal role for the first embodiment of IoT for cars. It may be worth mentioning that a wireless connection is not in itself sufficient to evoke the IoT paradigm. Considering vehicles as things, IoT can be considered glue between the vehicle and the outer world, offering, from case to case, a proper management and monitoring (abstraction) interface. Additionally, each car is expected to become more and more a cluster of things, rather than just a thing: in fact, in future, a plethora of on-board, mutually connected, devices are likely to be either integrated in the car or to accompany a driver.

In terms of the existing initiatives in the domain of the ”connected car”, they are still mostly fragmented within different countries and companies and funded by national governments, framework programs or individual companies in the automotive field.

The main Standards Developing Organizations, Alliances & Open Sources Initiatives (OSI) that are focusing significantly on Smart Mobility vertical industry domain are: CENELEC, Ca2Car Communication Consortium (C2C-CC), SAE International, CIA, ERTICO, Car Connectivity Consortium, ISO, Open Automotive Alliance (OAA), Genivi Alliance, Industrial Internet Consortium (IIC), AIOTI, IEEE, ETSI.

Figure 3.1: The connection between ETSI and the active development and stardandization projects, bodies and organizations.

The main SDO/Alliances/OSI that are focusing on the horizontal industry domain and can be as well applied in the Smart Mobility vertical industry domain are: OSGi Alliance, Hyper/CAT, IETF, Thread, ISO/IEC JTC1, BBF, W3C, ITU, GSMA, OneM2M, OASIS, WWRF, 3GPP, OMG, LoRA, eClass, Bluetooth, Weightless, OGC, ipSO Alliance, OMA, Zigbee Alliance, The OpenGROUP, Allseen Alliance.

To date, a number of use cases can be built on common standards and networking technologies:

- Wireless and automated payment can be done by the use of smartphones and smart cards with RFID and/or Near Field Communication and the Calypso electronic ticketing standard,

- Real-time road usage information exchange can be achieved using the DATEX II or CEN SIRI standards and the initiatives around the INSPIRE data specifications with TN-ITS/ROSATTE on ITS-related spatial data,

- Positioning vehicles and public bikes can be done by using GPS/Galileo devices,

- Automated road tolling can be achieved with technologies in the European road tolling standard, such as Dedicated Short Range Communication (DSCR), Automated Number Plate Recognition (ANPR) or the Global Navigation Satellite System (GNSS),

- Automated notification of emergency services can be done with the eCall (emergency call) standard,

- Cloud platform connectivity can be achieved with networking services, such as 3G/LTE, WiMAX, MobiquiThings or Sigfox/LoRa,

- Networking services towards vehicles can be achieved making use of the IEEE802.11p standard as a part of the G5 protocol suite standardized by ETSI,

- Parking spot availability (as well as infrastructure monitoring) can be checked using a wireless sensor network complying withIEEE802.15.4 technology for Physical and MAC layers, and IETF 6LoWPAN/CoAP for Networking and Facility layers,

- Vehicle to Vehicle (V2V) and Vehicle to Infrastructure (V2I) communication services can be achieved with technologies like 3GPP LTE-V2X (Vehicle Direct Communication through proximity services) or 802.11p.

As regards the software, there are many ways to develop services for vehicles:

- run apps in the in-vehicle entertainment systems (Blackberry QNX CAR, Windows Embedded Automotive, Automotive Grade Linux and Android, Tizen [6]),
- use a link to a smartphone (Airbiquity, OpenCar, CloudCar, SmartDeviceLink /AppLink, MirrorLink, Apple CarPlay, Google Open Automotive Alliance and Windows in the Car),
- remote access to the vehicle through an API (OnStar, General Motors API, Ford Remote API, Airbiquity, reverse engineering of vehicle protocols),
- access to data through the On Board Diagnostics port using the OBD-II diagnostics protocol (Dash Labs, Mojio, Carvoyant, MetroMile and smartdrive.io, OpenXC),
- run applications and services using On Board Units and Road Side Units belonging to Vehicular Ad Hoc Networks (VANETs) and communicating via IEEE 802.11p standards.

3.2 Intelligent Transport Systems

ITS are defined by ETSI in as systems "to support transportation of goods and humans with information and communication technologies in order to efficiently and safely use the transport infrastructure and transport means (cars, trains, planes, ships)".

They therefore span multiple environments and target a large range of applications within many operational fields.

Independently from the field and the environment they are embedded in the standards recognize two domains: the ITS domain, which is the set of technologies and entities collaborating with the C-ITS standards and the general domain, which involves all the technologies which refers to the application domain or other domains. These two domains compose the scenario in which the C-ITS entities are the deployed.

The last item refers to the so called Cooperative Intelligent Transport Systems (C-ITS), that will be discussed in details in the very next section.

3.3 Cooperative ITS

The most recent achievements of the activities in the area of intelligent transportation systems (ITS) promoted by academia, industrial stakeholders and Standard Development Organizations (SDO) are the so called Cooperative ITS (C-ITS) [7].

Their primary goal is to use and plan communication and sensor infrastructure to increase road safety. Cooperation on the road includes car-to-car and car-to-infrastructure communications. Data available from vehicles and road side units can be either consumed locally in the boundary of a geo-localized network or transmitted to a remote server for central fusion and processing. These data can then be used to detect events such as road works, traffic jams, and approaching emergency vehicle. This information can be processed in order to produce driving recommendation dedicated to a single or a specific group of drivers and transmitted wirelessly to vehicles.

3.4 Vehicular communications

Depending on the application, there are mainly two wireless technologies considered for C-ITS communications: the short-range communication technology IEEE 802.11p [8] and cellular networks such as 3G/LTE (and in the future, 5G). The latter depend on a centralized network topology where all data traffic must take a detour via the base station (BS), i.e. the operator's antennas and facilities, even though two stations are geographically co-located. IEEE 802.11p, on the other hand, offers the ability for direct communication between ITS stations, i.e., ad hoc communication, for up to 1000 meters.

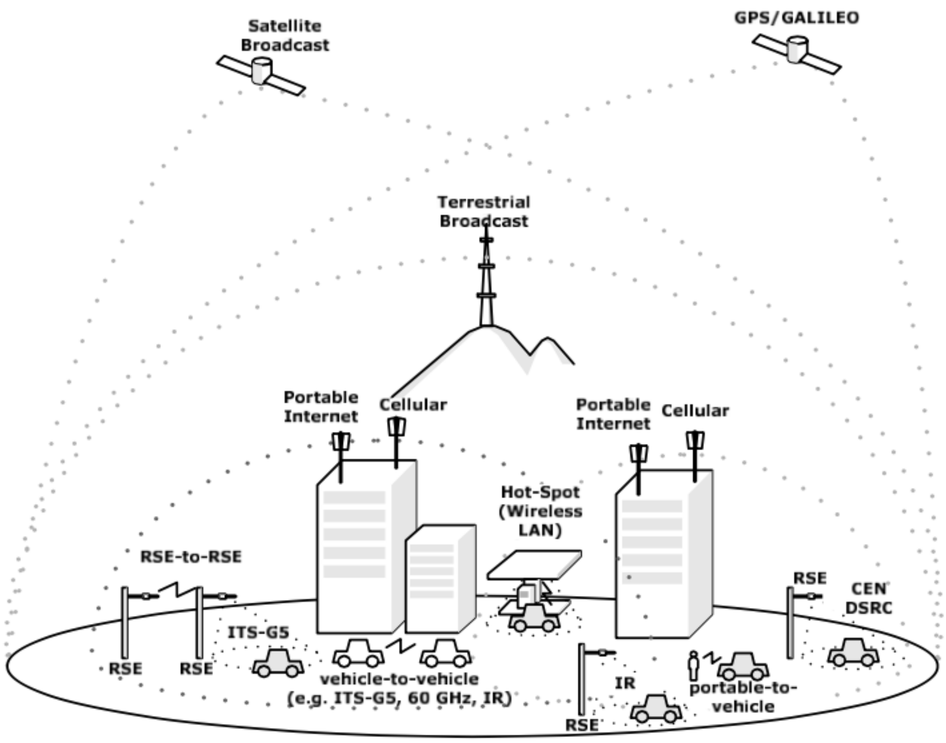

Figure 3.2: The envisioned ITS Scenario.

Given the different communication requirements of different C-ITS applications 3G/LTE and IEEE 802.11p are not two competing technologies but actually they rather complement each other. The peculiarity of 802.11p is capability of dissemination of information locally around the ITS station thus providing a communication architecture scalable with regards to the number of vehicles and reducing sensibly communication latencies.

Through the introduction of specific routing protocols, information can also be transmitted in a certain direction or to a certain geographical area (georouting). The information contained in messages generated by cars is perishable with short life-time due to the movement of vehicles: the higher speed of the vehicle the shorter time the information is valid (in particular this concerns position information) therefore, by using the ad hoc communication of IEEE 802.11p, delays can be kept low since no detour around a BS is necessary. C-ITS applications that do not have short delay requirements or rely on information being spread regionally rather than locally can utilize 3G/LTE, but this could be crucial otherwise. For example, a fixed road side infrastructure equipment using 802.11p can act as an information provider to services offered by 3G/LTE by transmitting service announcements to surrounding vehicles, which in turn can connect to the 3G/LTE network themselves to retrieve more information. It should be noted, however, that 3G/LTE networking currently requires subscription to a specific mobile telephone operator, which is not necessary for 802.11p used for C-ITS.

3.4.1 IEEE 802.11p

Entering the details, IEEE 802.11p is an amendment to the ubiquitous wireless local area network (WLAN) standard IEEE 802.11 tailored to the vehicular environment which, as we stated above, enables devices to communicate without having to associate with each.

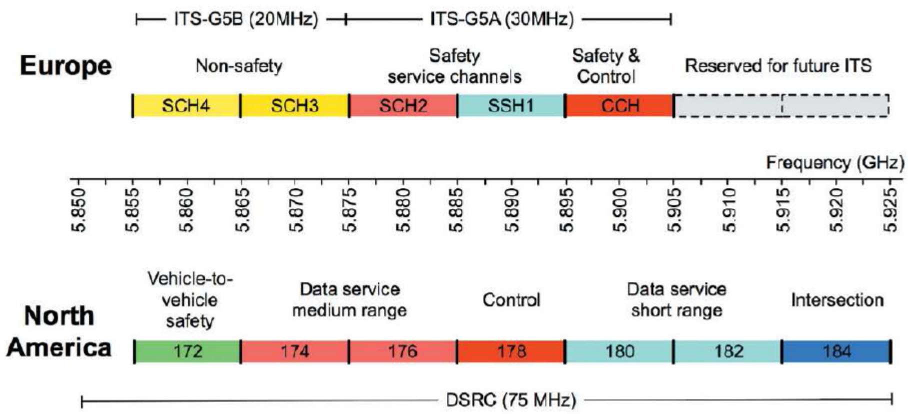

Figure 3.3: Public spectrum bandwidth and channel allocation for IEEE 802.11p.

Indeed, as the communication link between two peers might exist for only a short amount of time, the IEEE 802.11p amendment defines a way to exchange data through that link without the need to establish a basic service set (BSS), and thus, without the need to wait for the association and authentication procedures to complete before exchanging data. For that purpose, IEEE 802.11p conform stations use the wildcard BSSID (a value of all 1s) in the header of the frames they exchange, and may start sending and receiving data frames as soon as they arrive on the communication channel.

Because such stations are neither associated nor authenticated, the authentication and data confidentiality mechanisms provided by the IEEE 802.11 standard (and its amendments) cannot be used. These kinds of functionality must then be provided by higher network layers and/or by cross layer interfaces.

At the physical layer, IEEE802.11p uses the orthogonal frequency division multiplexing (OFDM) with channels of 10MHz bandwidth in the 5.9GHz band (5.850-5.925 GHz). The normative allocation bandwidth for Europe and North America are shown in Fig. 3.3. This is half the bandwidth, or double the transmission time for a specific data symbol, as used in 802.11a. This allows the receiver to better cope with the characteristics of the radio channel in vehicular communications environments, e.g. the signal echoes reflected from other cars or houses. Support of three transfer rates is mandatory: 3 Mbit/s, 6 Mbit/s, and 12 Mbit/s.

In Europe, 802.11p was used as the access layer of the ITS-G5 standard (G5), supporting the GeoNetworking protocol for vehicle to vehicle and vehicle to infrastructure communication [9].

Moreover, it is interesting to note that ETSI has reserved three channels (a band of 30 MHz at 5.875-5.905 GHz band) solely for *traffic safety* applications and other four channels (two bands of 20 MHz at 5.855-5.875 GHz and at 5.905 - 5.925 GHz) for non safety and future ITS applications.

3.5 The Basic Set of Applications

ITS systems are characterized by the mobility of the stations and high dynamics topologies, by any kind of communication technologies, by potential support of any kind of application support (for ITS-C, use of the station as a transparent communication means, station internal communications only).

Within the framework of ETSI standardization, the Basic set of application has been identified, which is the set of capabilities that are foreseen to be available in automotive systems within 2-3 years from the definitions of the standard.

A basic set of applications has been defined in [10] by ETSI TC ITS, which has been grouped into road safety, traffic efficiency, and other applications.

3.6 ITS Station

The main goal of this thesis was the creation of a system following the concepts and requirements of an Intelligent Transport Systems Station and as we will see, the concept of Station is fundamental within the architecture envisioned for these type of systems.

This concept, as we will see in the next sections, is able to accommodate and uniform many different implementation strategies and architectures. Indeed, starting from small dedicated MCUs deployed within vehicles now an ecosystem of devices is populating them, from mobility and mechanical control systems, to integrated diagnostic tools, to personal portable and wearable devices. The concept of ITS Station (and its categorization) can integrate all this different technology and most of all to interconnect them safely and efficiently, leaving space for un-countable new applications to be developed for safety, efficiency and infotainment.

From now on, when using the term "Station" we will refer to this aforementioned concept.

3.7 Station architecture

The concept of station represents a set of functionalities which forms the basic brick of any ITS system or subsystem. The comprised functionality are structured following the OSI layer model and their structure form the architecture of a station.

As the reader can see in Fig 3.4, with respect to the ISO/OSI communication layers the reference architecture for a station is as follows aggregates Layers 1 and 2 (PHY and MAC) into the “Access” layer and it aggregates Layers 3 and 4 (NET and TRANSPORT) into the “Network” layer.

On top of the Network layer, a layer called “Facilities” is introduced. This layer contains several functionalities some of which can be similar to the Presentation and Application (layers 5 and 6) of the ISO/OSI stack but also include many other functionalities, both application specific or common to many (or all) supported applications. We will describe the Facility layer in more details in section 3.8.

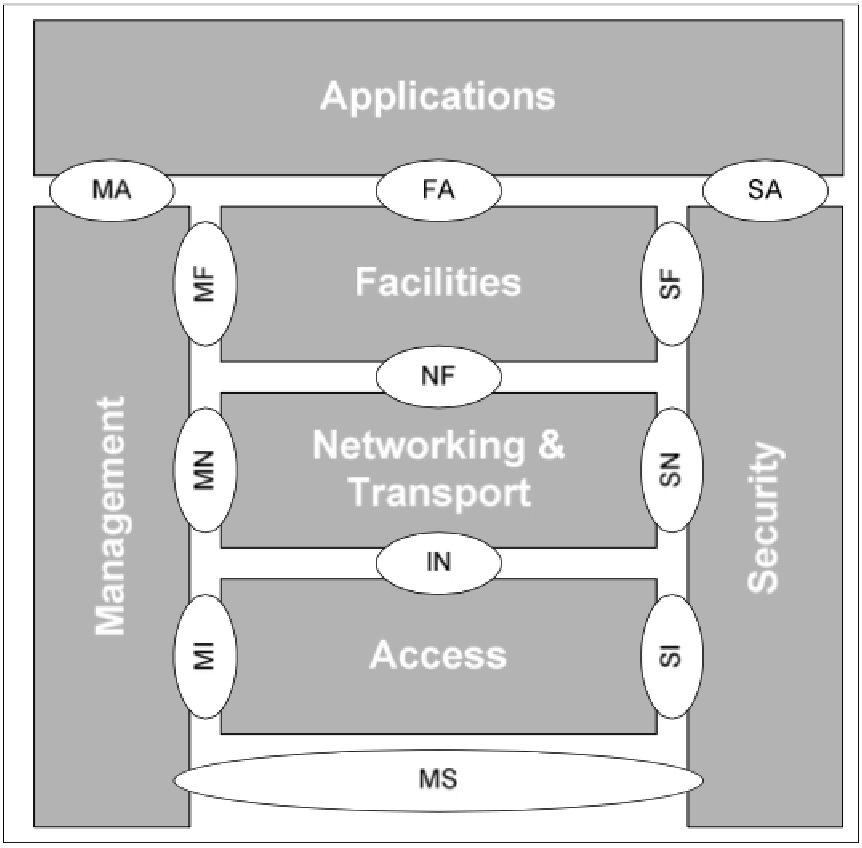

Figure 3.4: The reference architecture for the ETSI C-ITS Station.

Beside the above mentioned layers, two classes of services and functionalities are grouped into two entities: “Security” and “Management”. Differently from the design of the ISO/OSI stack, where each layer has its own management and security mechanisms, in the ITS Station architecture these two entities are unique and used by all the other layers to provide many non-functional services.

On top of the Security, Management and Facilities layers the “Application” layer is defined. The applications in a ITS Station are modules which make use of data, messages and other information provided by other layers. Therefore, also in this case, despite the same name, they should be distinguished from the Application layer of the ISO/OSI stack.

The reference architecture for the station is replicated in each entity composing an ITS system or subsystem. In particular each vehicle, road side infrastructure and remote control equipment should be seen as a set of stations, routers and gateway. Each of these contains itself (part of) the reference architecture.

3.7.1 Different types of stations

At the beginning of this section, we stated that the concept of Station is able to intercept many different entities in the entire ITS scenario. Here we show a categorization that help two understand how many different tasks and functionalities can be assigned to a Station and its sub-systems. First, following the technical specification, we categorize ITS Stations by means of the segment of deployment in the ITS scenario:

- a *Central* ITS-S is deployed in the remote data-center (w.r.t. the road environment) and is in charge of monitoring, data fusion, remote service provider;
- a *Road side* ITS-S is deployed on equipment distributed around the road environment, e.g. smart traffic lights;
- a *Vehicle* ITS-S is on board of any kind of vehicle (car, bus, truck, train) and could be delegated to mechanical control, communication, infotainment;
- a *Personal* ITS-S embedded in personal, portable devices as tablets, smart phone, watches or any other wearable device.

Also, we can try to better define the kind of functionalities that could be implemented in a station by defining the functional sub-systems of an ITS station (shown in Fig. 3.5):

- ITS-S host,
- ITS-S gateway,
- ITS-S router,
- ITS-S border router.

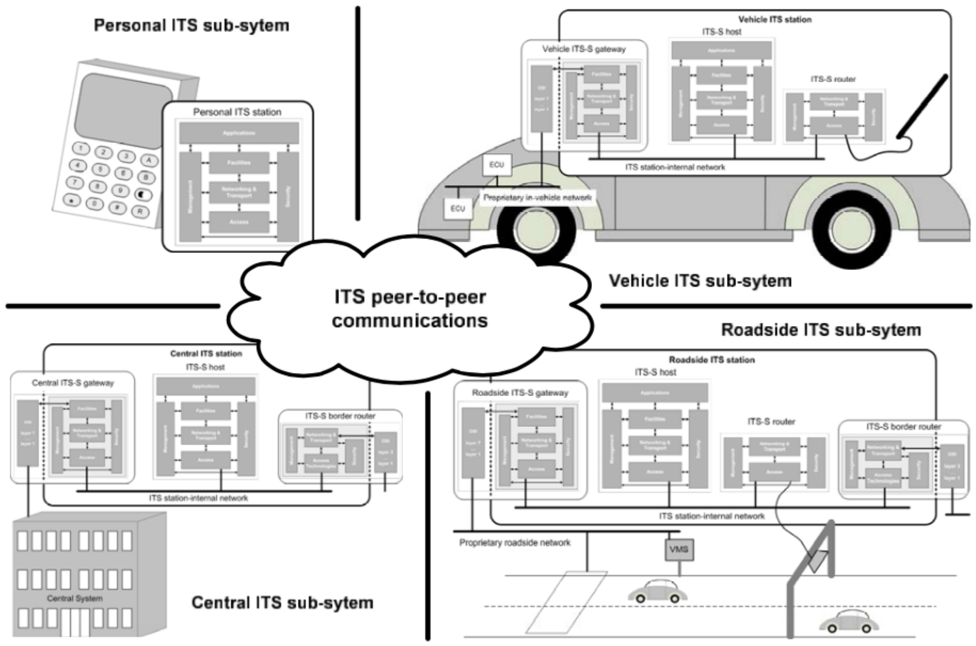

Figure 3.5: Different types of ITS-Stations and their respective sub-systems.

The gateway, router or border router can be generally called ITS-S *interceptors.*

Generally, the ITS station is said to connect to access networks via ITS-S border routers and it connects to ITS ad hoc networks via ITS-S routers. The internal network is not necessarily physically available and may be realized simultaneously with different access technologies, both wired, e.g. Ethernet, or wireless, e.g. BlueTooth. The ITS station typically connects to a proprietary network via an ITS-S gateway, i.e. the central ITS-S gateway, the vehicle ITS-S gateway and the roadside ITS-S gateway but the approved standards do not restrict any specific implementation approach.

3.7.2 Application layer overview

An ITS application is an association of two or more complementary ITS-S applications, e.g. server part and client part. ITS-S applications reside in the "Applications" layer of the ITS station reference architecture. An application makes use of the underlying facilities and communication capacities provided by the ITS-S. ITS applications are divided, within the standards, in three classes: road safety, traffic efficiency and other applications.

C-ITS services shall be designed to support multiple classes of ITS applications including those supporting vehicle operations. Dependent on how much these applications rely on the communication services, application classes impose more or less stringent requirements on C-ITS, with respect of reliability, security, latency, and other performance parameters.

All ITS applications shall be associated with an ITS application priority according to the functional and operational requirements of the ITS application. This ITS application priority shall indicate the maximum possible value of the channel access priority. They also shall be associated with a specific logical channel type for transmission of data packets

Maintenance of ITS applications, i.e. installation, de-installation, updates, shall be performed in a secure way in order to support protection of ITS stations from attacks by malicious applications and the facilities layer provides the required functionalities based on interactions with the security entity.

3.7.3 Access: C-ITS OSI protocol stack

In the field of ITS many wireless communication technologies are used: Cellular based (3G, LTE, 5G, WiMAX), Dedicated Short Range Communication (CEN DSRC 5.8 GHz), Vehicular Communications (WAVE/CALM-M5/IEEE 802.11p 5.9 GHz) and Infrared (CALM-IR), while Wireless Personal Area Network (WPAN).

GeoNetworking

The requirements at network layer that are more specific to the dynamics of the vehicular networks are met, in the ETSI ITS architecture, GeoNetworking set of standards. Nonetheless, the standards include support for Internet access through IPv6 and TCP/UDP (not excluding other transport protocols).

The two central GeoNetworking standards in the network layer are divided into: (i) media-dependent functionalities and (ii) media-independent functionalities. The former is specified in TS 102 636-4-1 and deals with the Decentralized Congestion Control (DCC) support on the network layer specifically tailored towards the access technology ITS-G5. The media-independent functionalities are found in TS 102 636-4-2 and it specifies packet types that can be used in the different GeoNetworking scenarios. It is also possible to transmit IPv6 datagrams over the GeoNetworking protocols and this is described in TS 102 636-6-1.

The GeoNetworking protocol packet consists of two parts: the mandatory common header and the optional extended header. The common header is fixed to 36 bytes and it contains amongst other things geographical information about the sending station, which is the majority of the header length (28 bytes). The extended header part varies between 0-60 bytes depending on communication scenario. The GeoAnycast and GeoBroadcast communication scenarios add the longest extended header - 60 bytes.

Within the GeoNetworking protocols, point-to-point communication (unicast communication) is supported, where the packet needs to be relayed through intermediate stations when transmitted from source (transmitter) to destination (receiver), multi-hop communications.

Also Point-to-multipoint communication is envisioned where more than one destination is interested in receiving the information transmitted. Point-to-multipoint is broadcast communication.

A novel concept introduced in GeoNetworking is GeoAnycast communication. It defines a geographical area of interest in which the information can be received by any station within the area. The sender is located possibly outside the geographically interesting region and there may be one or several stations relaying the packet in-between. In GeoBroadcast communication instead, a geographical area of interest is also defined, and when the packet reaches the destination area it will be broadcasted within the area.

Basic Transport Protocol

The Basic Transport Protocol (BTP) provides an end-to-end, connection-less transport service in the ITS ad hoc network. Its main purpose is the multiplexing of messages from different processes at the ITS Facilities layer, e.g. CAM and DEN services, for the transmission of packets via the GeoNetworking protocol as well as the de-multiplexing at the destination. BTP enables the protocol entities at the ITS Facilities layer to access services of the GeoNetworking protocol and to pass protocol control information between the ITS Facilities layer and the GeoNetworking protocol.

Message multiplexing/demultiplexing is based on ports, an ITS station-internal, 16 bit address.

BTP is a lightweight protocol: It has a 4-byte protocol header and requires minimal processing. It provides an unreliable transport of packets, i.e. packets can arrive out-of-order, appear duplicated or can be lost. The design of BTP assumes that entities using the protocol are either tolerant against the unreliable packet transport or provide appropriate mechanisms for reliable communication in their protocols.

3.8 Facility Layer

The facility layer is a middleware composed of multiple facilities. A facility is a component that provides functions, information or services to the ITS applications. It exchanges that are lower layers and with management and security entities of the ITS-S.

3.8.1 External gateways

They provide gateway functions for these external systems to exchange info with facility layers of other ITS-S.

Gateway to in-vehicle network

Facilities and applications receive from this gateway the required in-vehicle data in order to construct messages (mostly CAM and DENM) and for applications custom needs.

The specifications of gateway within the standard did not define in details the structure and functionalities because most of them rely on proprietary networks. In general we can say that, following a REST-ful paradigm, a gateway facility should have at least "*get*" and "*set*" operations towards resources of the in-vehicle network.

Road side gateway to central ITS-S

Central ITS stations could be the civic traffic monitoring center or the auto vendor data center or a third party application data center. In particular, RSUs deployed in a smart city would provide data to the city monitoring center but could be managed by the third party provider of the devices or of the connectivity.

From RSU or Central to on road equipment

Enables transmission of packets from/to road side equipment e.g. traffic lights, smart cameras, sensors. Here the IoT stack can be beneficial to reduce operational, infrastructure and management costs.

3.8.2 Facilities layer functional architecture

The general functional architecture of the facilities layer is illustrated. A set of facilities are identified in order to support the Basic Set of Applications. These facilities can be categorized according to two different approaches.

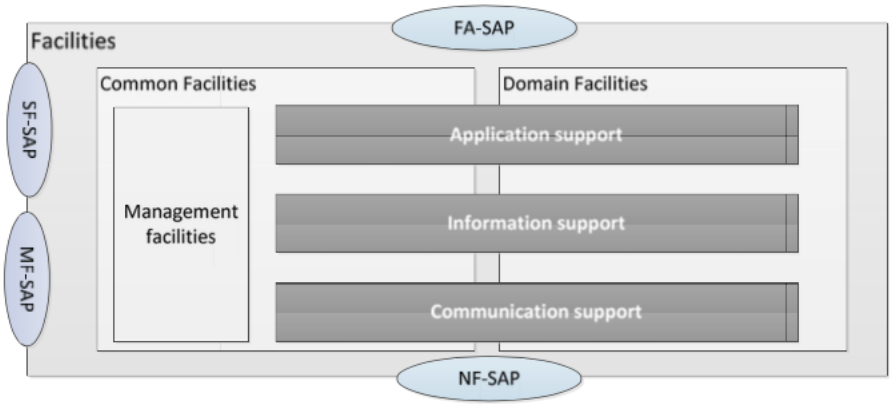

Figure 3.6: Block diagram of the internal functionalities and interfaces of the Facility layer.

The first approach distinguishes the facilities according to their main users: therefore we will have a set of *common* facilities, which are the ones that implement basic core services to support reliable operations, e.g. time and position services, and, on the other hand the set of *domain* facilities which provide functionalities for one or several BSA applications. Some of these facilities are optional for the station depending for example on the station type.

The second approach distinguishes the facilities according to the type of functionalities supported:

- Application support
- Information support
- Communication support
- Management support

The block diagram of the internal structure of the Facility layer is shown in Fig.3.6. The Service Access Points shown in the figure, are logical channels that enable full-duplex communication among layers. The list of common facilities (basic core services to support reliable operation of the station) is given in Tab. 3.1.

Table 3.1: The list of common facilities.

Classification	Facility name
Management	Traffic class management ITS-S ID management AID management Security access
Application support	HMI support Time service Application/facilities status management SAM processing
Information support	Station type/capabilities ITS-S positioning service Location referencing Common data dictionary Data presentation
Communication support	Addressing mode Congestion control

The list of domain facilities (services and functions for one or several specific BSA applications) is given in Tab. 3.2.

Table 3.2: The list of domain facilities.

Classification	Facility name
Application support	CA basic service DEN basic service EFCD Billing and payment SPAT basic service (signal phase) TOPO basic service (road topology) IVS basic service (in vehicle signage) Community service user management
Information support	Local Dynamic map RSU management and communication Map service
Communication support	Session support Web service support Messaging support E2E geocasting

3.8.3 Cooperative Awareness

We now introduce a very significant example of a ITS facility, the Cooperative Awareness facility, for several of reasons.

First of all this facility supports some of the most promising and characteristic applications which are the Road Safety applications. The facility is based on the ability of transport entities (vehicles, roadside units, pedestrian, etc.) to collect information useful to create knowledge in order to make more intelligent use of the transport infrastructure.

Moreover this facility is one of the peculiarity in ITS stack and (together with the Decentralized Event Notification facility) one of the few new protocols at the application layer tailored for vehicular network needs. Finally, the Cooperative Awareness (CA) facility will be one of the tools that we used to integrate the IoT within automotive systems, therefore a more detailed notion of it will clarify our integration.

The main purpose of the Cooperative Awareness facility is to share information about the status of the ITS and to collect information about as many objects in the environment as possible (e.g. other vehicles, pedestrians, road side unit, etc.). This facility is envisioned to support primarily safety applications (and notably Collision Detection and Avoidance) but also to feed a database of "near" objects known as Local Dynamic Map (the creation and management of which is demanded to another specific facility). The possible collision risk computation and collision avoidance applications are capable of getting detailed real time data about the environment and apply countermeasures through HMI notifications towards the driver or autonomous prevention mechanisms. The CA facility provides functions for the management of some sort of heartbeat messages called CA messages (CAMs) and operates the CAM protocol. Cooperative Awareness Messages contain various information about the status, attribute and capability of the originating ITS-S. Operating the CAM protocol means actually supporting three operation: sending, receiving and disseminating CAMs.

Sending CAMs comprises the generation and transmission of CAMs. The dissemination, after being triggered by the CA facility, is actually operated by the Network and Transport layer. In a ITS-G5 A network CAMs are sent to all the ITS-Ss within the direct communicating range (one hop) but this communication range can, actually, vary on the transmission power selected by the ITS-S itself.

The CA facility resides, according to the standards, in both OBU and RSU and its duty is to constructs and transmits CAM at variable intervals. Transmission interval is set to be between several hundred milliseconds to one second, i.e. between $10Hz$ and $1Hz$. It may vary depending on the network status, e.g. congestion, but also on the status of the vehicle, e.g. position, speed.

The facility also receives CAMs, decodes them and provides the information to a facilities layer database Local Dynamic Map and/or to the ITS applications.

Functional description and requirements

Every ITS-S taking part in the road traffic shall disseminate CAMs (vehicle, personal, road side..) in order to offer the complete knowledge as most as possible to the ITS-S present in local area.

CA basic service shall communicate with other entities belonging to the

ITS-S to collect the data to be inserted in the CAM. Potential entities for data collection in a vehicle ITS-S may be the Vehicle Data Provider (VDP) and the Position and Time management (POTI) and the Local Dynamic Map (LDM) which collects the data coming from other stations.

The interface exposed by the CA facility should allow the following operations:

- Encode CAM: constructing and encoding the CAM. The most recent in-vehicle data shall be included in CAM.
- Decode CAM: decoding the received CAM
- CAM transmission management: implementing the protocol operation of the originating ITS-S and in particular:
  - Activation and termination of CAM transmission operation
  - Determination of the CAM generation frequency
  - Trigger the generation of CAM
- CAM reception management, which means:
  - triggering the "decode CAM" function at the reception of CAM
  - Provisioning of the received CAM data to LDM or ITS applications of the receiving ITS-S
  - Optionally, checking the information of received CAMs

Interaction with other facilities

The CA facility should of course be able to exchange commands and data with other facilities and applications through the interface to data provision facilities, i.e. facility layer entities that provides the required data to construct a CAM.

At the originating ITS-S, in order to actually propagate messages onto the network, the CA basic service should also provide the CAM to the Networking layer and all the metadata necessary for a correct utilization of network protocols. In order to do so, the standard defines Facility layer service data units (FL-SDU) which are data structure containing the actual CAM data (the payload) together with protocol control information (PCI) to properly set up the packet header (e.g. source and destination ports). At the receiving ITS-S the ITS N&T layer will pass the received CAM to the CA basic service, if available.

3.8.4 Message packet encapsulation

The interface between the CA basic service and the NT layer relies on the services of the GeoNetworkng/BTP stack or on the IPv6 stack and the combined IPv6/GeoNetworking stack.

In case the BTP/GeoNet stack is used, since the CAM should be sent to all ITS-S in communication range with no forwarding, the GN packet transport type Single-Hop Broadcasting should be used.

Table 3.3: GeoNet/BTP Protocol Control Information that should be passed from the CA facility to the NT layer

BTP/GeoNet option	Required value
Packet Type	BTP Type header B
Destination port	*Configurable*
Destination port info	*Configurable*
Packet transport type	GN Single-Hop Broadcast
Communication profile	ITS-G5
Security profile	both secured or unsecured possible
Maximum packet lifetime	*Shall not exceed 1000 ms*
Length	*Actual length of the CAM*

Table 3.3 shows the relevant requirements of the content of PCI being passed from CA basic service to the GN/BTP stack. A CAM may also use the IPv6 stack or the combined IPv6/GeoNet stack for dissemination and the transmission interface can be the same to any IPv6 service stack.

CAM dissemination algorithm

As we have seen, it is required that point to multipoint communication shall be used for transmitting CAMs. When ITS G5 is used for CAM dissemination, the control channel (G5 CCH) shall be used and the CAM shall be transmitted only from the originating ITS-S in a single hop to the receiving ITS-S located in the direct communication range of the originating ITS-S. Therefore CAMs are never forwarded.

For vehicle ITS-S the CA basic service shall be activated with the ITS-S activation and shall be terminated when the ITS-S is deactivated while for other types of stations it is not specified.

Generation is managed by the CA basic service. The service defines the time interval between two consecutive CAM generations. This interval can assume values between 100ms and 1000ms as we stated above, i.e. the CAM generation rate is between 1Hz and 10Hz.

To describe how the generation interval is computed, let be $T\_GenCam\_Min = 100m$, $T\_GenCam\_Max = 1000ms$ (the minimum and maximum values for the interval). The generation shall be checked repeatedly every $T\_CheckCam\_Gen$, which should be equal or less than T_GenCam_Min.

Now we introduce a fifth parameter, T_GenCam_Dcc, which provides the minimum time interval between two consecutive CAM generations in order to reduce the CAM generation according to the channel usage requirements of the Decentralized Congestion Control (DCC), which is Management functionality. The value of this parameter is therefore provided by the Management entity and will depend on Configuration and various Management policies. This value is limited by

$$T\_GenCam\_Mim \leq T\_GenCam\_Dcc \leq T\_GenCam\_Max$$

T_GenCam is the actual parameter that indicates the currently valid upper limit of the CAM generation interval.

Therefore let us list the condition that trigger the propagation of a new CA message:

1. The time elapsed since the last CAM generation is equal to or greater than T_GenCam_Dcc *and* one of the following ITS-S dynamics related condition is given:

    - the heading of the vehicle has changed by more than $4°$,
    - the position has changed by at least $4m$, or

- the speed has changed by more than 0,5 m/s.

2. The time elapsed since the last CAM generation is equal or greater than T_GenCam *and* equal or greater than T_GenCam_DCC.

On any of the above cases a CAM propagation shall be granted immediately.

In case condition (1) is verified, T_GenCam is set to the time elapsed since the last CAM generation. In case of condition (2) the parameter shall be set to T_GenCam_Max, after triggering N_GenCam consecutive CAMs. The parameter N_GenCam can be dynamically adjusted according to some environmental conditions. The default value shall be 3.

CAM format specification

A CAM is composed by one common ITS PDU header, which includes the information of the protocol version, the message type and the ITS-S ID, and multiple *containers*.

For vehicle Stations, CAM shall comprise one basic container and one high frequency container, and may also include one low frequency container and one or more special containers. All CAM generated by a RSU ITS-S, instead, shall include a basic container and optionally more containers.

The *basic container* includes basic information related to the originating ITS-S, i.e. the type of originating ITS-S (OBU, RSU, ...) and the latest geographic position of the originating ITS-S as obtained by the CA basic service. It shall be present for CAM generated by all ITS-S implementing the CA basic service.

The *high frequency* container contains highly dynamic information of the originating ITS-S, such as speed of heading.

The *low frequency* container contains static and not highly changing dynamic information, like status of exterior lights.

The *special container* contains information specific to the vehicle role of the originating vehicle. This can be used for vehicles with particular roles, such as emergency vehicles and public transport which should provide additional information.

Each container is composed of a sequence of optional or mandatory data elements and/or data frames which are described in ASN.1 format within the technical specification. The semantic of the various fields is described in the Common Dictionary facility.

Other time requirements

The low frequency container shall be included and transmitted with a rate of no more than 2Hz. The same applies for the special vehicle container. They are both included in the very first CAM transmitted from the CA service activation.

For RSU, the maximum transmission rate of CAMs should be 1Hz but it should be high enough to let any vehicle to receive at least one CAM originated by the RSU when it is within the communication zone.

Each CAM shall be timestamped according to an acceptable time synchronization.

Time required for a CAM generation should be less than 50ms. The time required for a CAM generation refers to the time difference between time at which CAM generation is triggered and time at which the CAM is delivered to N&T layer.

The timestamp used for CAMs should correspond to the time at which the reference position of the originating ITS-S given in this CAM was determined. The format and range of the timestamp should be as follows: the value should be wrapped to 65536. This value is set as the remainder of the corresponding value of `TimeStampIts` divided by 65536 as below:

$$timestamp = TimeStampIts \ \% \ 65536$$

`TimeStampIts` represents an integer value in milliseconds since 2004-01-01T00:00:00:000Z as defined in ETSI TS 102 894-2. In the RSU the timestamp given in the CAM should be the time of generation. The difference between CAM generation time and time stamp shall be less than 32 767 ms.

Other requirements are set on security and fields value confidence, but we will omit them, since even though security deserves special attention, it is out of the scope of the present project.

3.9 Related work in recent European projects

Recently many R&D initiatives in the scope of "Connectivity and Communication" focus on the design and development of a fully networked car integrated with an Intelligent Transport System.

A number of prototypes implementing vehicular communication stack based on IEEE 802.11p have been proposed. Unless high-quality multimedia transmissions are required, the bandwidth results indicate that the data rate required by most of the traffic efficiency and comfort services can be covered. According to latency tests, even non-critical security services, which are not highly dependent on real-time response, could be implemented, such as emergency assistance, variable traffic signaling, automotive surveillance applications etc. (e.g. [11]).

One further step is reached by the integration of IPv6 in the ITS, which is part of the coexistence strategy to manage the heterogeneity of the involved technologies and architectures, in order to meet the interoperability across business, service providers, and users. One particular example is the ITSSv6 project [12], funded by a FP7 EC grant which has recently concluded with the development of a communication stack compliant with current ISO and ETSI standards in cooperative ITS. It is particularly focused on the usage of IPv6 as a common protocol for vehicular communication nodes. This approach is clearly beneficial for the inclusion of IoT communication patters, given the global connectivity of nodes, even the embedded ones. ITSSv6 demonstrated that IoT devices can be integrated into the ITS Station Reference Architecture defined by ISO/ETSI, which allowed ITS stations to communicate with low-power sensor networks over IPv6 (6LoWPAN). This was successfully demonstrated, along with several other ITS communication technologies such as IEEE 802.11p and 3G [13].

In the same line, the FOTsis project [14], which is being finalized at the moment, has ported the communication stack of ITSSv6 to create a communication architecture appropriate for C-ITS services in the short term. It is, with no doubt, a key proof of concept of the guidelines given by ISO and ETSI regarding the reference ITS communication architecture. In contrast with other international initiatives, FOTsis bet on using well-known IETF protocols in the ITS sector, embracing the forthcoming all-IP world and the integration of vehicle subsystems in the IoT expansion to consumer devices.

A step forward was achieved by the ICSI project [15], who demonstrated the benefits of a seamless integration of IoT devices into C-ITS infrastructure. ICSI promoted a new C-ITS architecture where the intelligence for sensing and actuation is distributed over some of the elements, called gateways, which host a software platform for running ITS applications, using the local storage and computation capabilities available. Among the achievement of ICSI projects, it is worth to recall the design and the realization of a WSN node that embeds a ITS station with the needed software suite including 6LoWPAN/CoAP protocols [1] and M2M Middleware. The latter is a software capable of supporting standard Machine-to-Machine communication for the ITS use-case, while tackling with the dynamic nature of resource-constrained devices and networks [16].

Hence, at the network layer, IPv6 is found the cornerstone for sticking two independent and huge partitions of the new era of the information and communication technologies: the Internet 2.0 and the car.

In the device domain there is still the need to demonstrate how to achieve a multi-interface system able to communicate with ad-hoc sensors; vehicular network and wideband backbone (3G/4G), maintaining interoperability features.

Chapter 4

Prototype system for IoT enhanced C-ITS services

4.1 Overview of the architecture

Starting from mature standardized technologies from the fields of IoT, automotive and cooperative ITS that we briefly presented in the previous chapters, we designed a general purpose platform capable of delivering non safety-critical services to a set of final users including other machines. In this chapter we will present our proposal of integration starting from a high level architectural interface and then presenting the actual implementation we provided, both in hardware and software components. As represented in Figure 4.1, the platform integrates many devices embedding suitable communication stacks, networking interfaces and high level software applications.

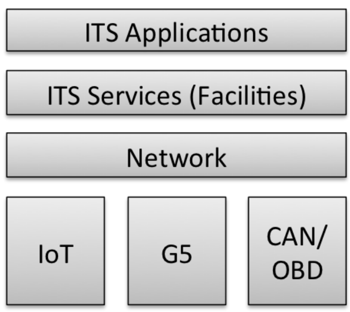

Figure 4.1: IoT/C-ITS system overview view

Regarding the communication components, we have considered the IoT stack following the specifications of the protocols presented in section 2.3 coming from IETF 6Lo Working Group in the area of Low-Rate Wireless Area Networks and from IETF CoRE Working Group for constrained devices application protocols, commonly known as 6LoWPAN and CoAP suite of standards. To properly support the functionalities envisioned for the ITS-S station as presented in section 3.6, in particular the communication toward the on board network for vehicular stations, we have considered the mandatory public interface of vehicle CAN bus (i.e. eOBDII [17]); finally, considering its importance as the specific vehicular communication medium, we have considered the IEEE802.11p access layer in compliance with ETSI specifications for the G5 communications illustrated in section 3.4.

Following the specifications coming from ISO/ETSI TC ITS standards in force, our goal was to design a platform as an ITS station having multiple concurrent access technologies integrated through dedicated network adaptors (see Figure 4.2).

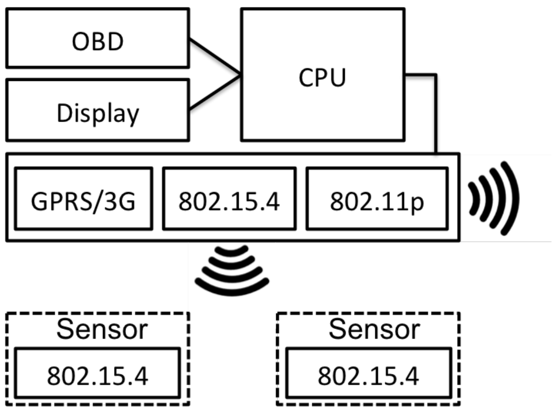

Figure 4.2: Architecture of the IoT/C-ITS station connecting to a WSN

We presented the Cooperative Awareness facility standardized by ETSI in the previous chapter, as the appropriate, standard compliant solution for the communication of non-reserved and non-sensitive vehicle information. CAMs are composed by different containers (following the common ITS PDU header) having useful information related to specifics object. The flexibility to add or to integrate one of the container (identified in accordance with ETSI specifications) with a section (or some elements) specifically dedicated to the "smart city features" has been exploited. In particular the "Special vehicle" container is intended to carry data for vehicles like ambulances, trucks, police, etc., so it represents the entity best suited to be expanded in order to include data for a new kind of special vehicles such as courier vehicles.

Indeed CAM special containers are the proper way to disseminate on the vehicular networks (both internal and remote) the information from the smart objects within the Station system. For example, in a Smart City scenarios, data related to the cargo status of vehicles or to a parking slot status have to be considered in order to implement services and policies for assisting the courier driver in joining the dedicated parking slots.

The architectural overview we described above was the topic of a scientific paper by *Carignani et al.* [18] that has been accepted to be included in the proceedings of the 2nd IEEE World Forum on IoT to be held in 2015, where the paper will be presented.

4.2 Design choices for the ITS Station

The main focus of this thesis was to design the ITS station in order to follow the ETSI standards guidelines in the vision to reach standard compliancy in the future while empowering our design with the capability to transparently integrate with the IoT.

Full compliance to the standard is two-folded: it calls for completeness and adherence. Completeness requires the implementation of all the entities envisioned within the architecture, that is the layers, security and management entities and within them of all the contained components. For example, within the facility layer, all the common facilities should be implemented. Adherence requires that all the contained components should be implemented in such a way that their functional and non-functional behavior respects the guidelines and requirements of the specifications. This means that all the components should do what is required and do it the right way.

Because of time resources, full compliance was actually not reachable within the project developed for the thesis, therefore we focused mostly on reaching functional objectives to create a real prototype (we recall that these matters are still research and innovation topics) to show the feasibility of the entire system. We therefore had to make some choices on which parts of the standard to implement. We will present in this chapter which features of the ITS-Station concept were developed and how, pointing when they meet the requirements from the specification and when they do not.

On the other hand, all hardware requirements to support a full connected ITS Station could be achieved and in particular we reached the requirements for the URBELOG project (implementing loading zone applications from the BSA for efficient last mile logistics), which was an opportunity to test on field the feasibility of our performances. We will discuss the project and its implementation in details in Chapter 5.

Moreover, the underlaying software architecture, based on the tailored framework we designed, to our prototype is structured in such a way it can be easily extended in order to accommodate further services, preserving all the requirements of separation, encapsulation, management and performance set by the standards for ITS Stations. We will describe the framework and its feature in section 4.4.2.

4.2.1 Hardware

Given the requirements for the URBELOG project, the target hardware to be used for development was previously designed by the Network of Embedded Systems at CNIT/Scuola Sant'Anna in collaboration with New Generation Sensors (NGS) s.r.l [19].

4.2.2 Communication interfaces

The hardware requirements for the station to be developed were many. First of all, the device required several interfaces, to support communication with the different components in the systems and to open possibilities for future integrations. The selected interfaces are :

- VANET (802.11p) - the transceiver for the vehicular network, enabling the station to communicate with other ITS-Stations;

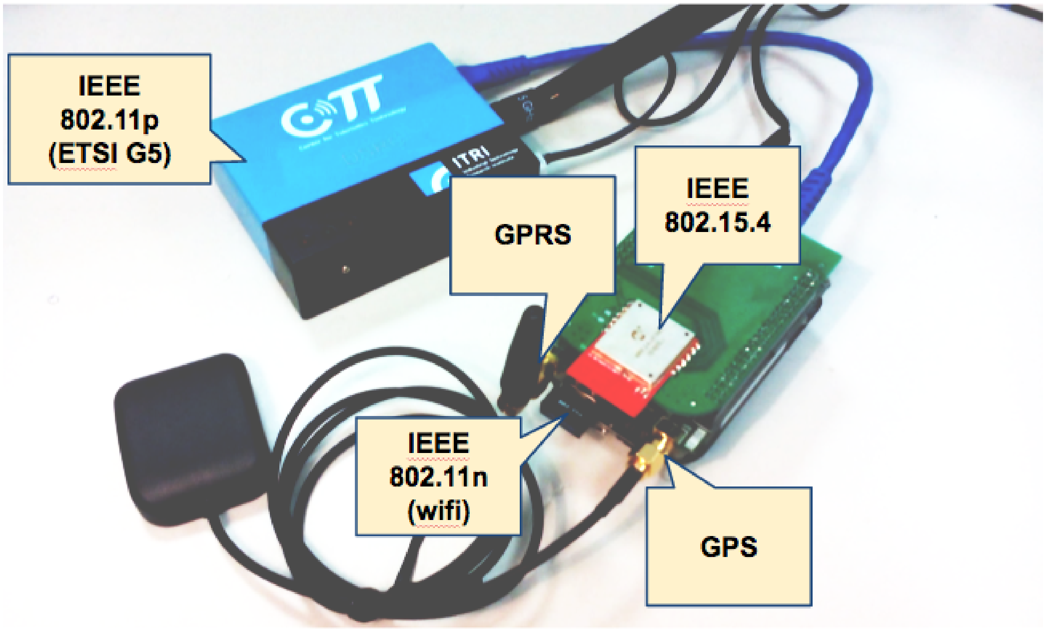

Figure 4.3: The selected hardware components and the integrated communication interfaces.

- 802.15.4 - the transceiver for the WSN/IoT stack, enabling the communication with on-board or road side wireless sensor networks;
- WiFi (802.11) - for management and, in the future, to connect wireless HMI devices (smartphones, tablets, vehicle equipment);
- Mobile (3G/4G) - for mobile Internet access, enabling the communication with the Smart City Center in RSUs and with the Vehicle Manager for the OBUs.

Moreover, two particular extensions were needed:

- CAN / OBDII - used only in vehicular ITS-S, used to collect information about the vehicle (speed, RPM, odometer, etc.)
- GPS - To enable the positioning service.

Other important requirements for hardware were compactness, low-cost, low-power consumption because the station could be deployed on vehicles and road side infrastructure.

The choice was done upon the Beagle Bone Black (BBB) development board to host the main logic of the station and to provide Ethernet, Wifi, Mobile, and CAN (for on board units) interfaces. The Beagle Bone Black is a development board based on the Texas Instruments AM335x 1GHz ARM Cortex-A8 processor, featuring 512MB DDR3 RAM, 4GB 8-bit eMMC on-board flash storage, 3D graphics accelerator.

For the IEEE 802.11p transceiver, to overcame the lack availability on the market of embeddable chipset to include onto the development board, a different board by ITRI, the IWCUv5, was attached to the system to act as a gateway toward vehicular networks. The IWCUv5 board is a standalone device designed to serve as a vehicular station itself. It is therefore able to communicate over the VANET specific protocols. In particular, it features a IEEE 802.11p 5.9 GHz transceiver and it implements the Link layer functionalities, the GeoNetworking network functionalities and the Basic Transport Protocol transport functionalities.

In the actual hardware implementation, the IWCUv5 works as a gateway from the Beagle Bone Black to a GeoNetworking network and viceversa. The two devices are connected over Ethernet.

The BBB board itself, features the Ethernet communication interface off-the-shelf and has been extended with:

- a USB WiFi (IEEE 802.11a/b/g/n) dongle,
- the TELIT GE864-GPS cape with GPRS and GPS receivers,
- the NGS cape with 802.15.4 transceiver from Microchip (the well known MRF24J40 [20]),
- CAN / OBD interface.

4.3 Software architecture

4.3.1 The station as a set of applications and services

In order to implement efficient and scalable applications on an ETSI compliant ITS station, we synthesized the design of the station as an orchestrator of services and applications.

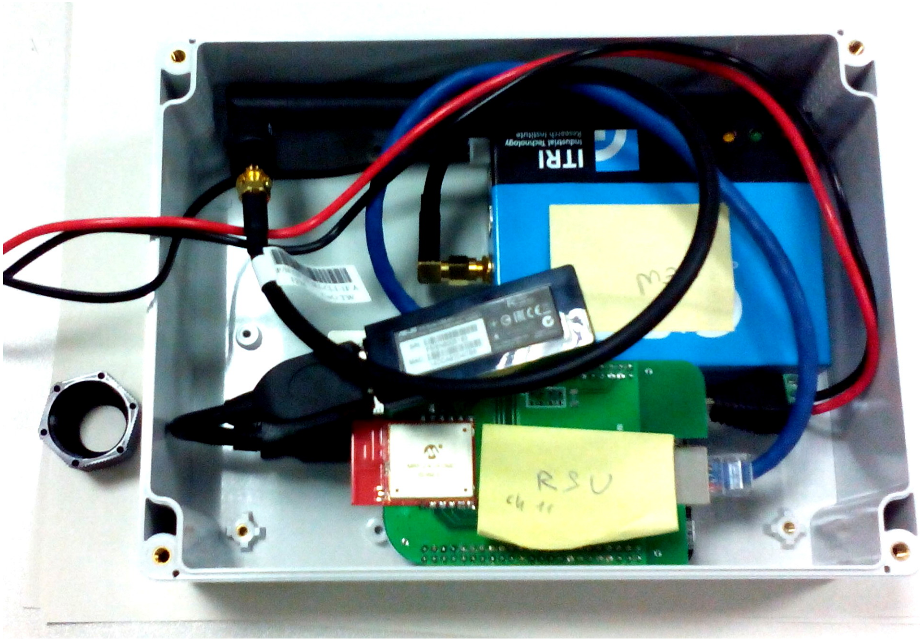

Figure 4.4: The hardware prototype set-up within a box for outdoor installation.

Due to the constrained resources and the vision to possibly deploy the station to even more constrained devices, one of the given requirements for the orchestrator was to be implemented in an shared-memory, object-oriented fashion, using a compiled language. C/C++ has been chosen due to the availability of compilers and presence of many libraries.

Different choices were made in other projects. For example, in the OVERSEE project the focus was given to isolation and protection against malicious or defected applications or services. This has led to the development of virtualized solutions: a bare-metal hypervisor is able to fully separate programs produced by different parties with the additional benefit of supporting both real-time and non-critical services with appropriate scheduling policies.

Our approach is different but not at all in contrast with this vision: the applications and services built upon the framework developed for this thesis could be easily deployed in any virtualized environment.

4.3.2 Access, Network and Transport layers

Since the availability of Linux as the operating system, the Access and Network and Transport layers functionalities of the station are mostly implemented in the kernel itself, in particular the ISO/OSI TPC/IP stack.

Concerning the GeoNetworking network, the usage of a separate board as a gateway required the development and the implementation of a library to send and receive messages and meta information above them.

Access to the GeoNet/BTP network

Since the BTP protocol provides end-to-end connectionless communications, the IWCUv5 runs a gateway daemon listening on a UDP port for messages and the information to fill the packet header. On the other way, when a message is received onto the network its payload is demultiplexed and it is sent over UDP to the BBB board where the orchestrator is listening to receive it.

Access to the IoT/WSN network

Another not natively supported networking stack is the IoT/WSN stack. Although a native Linux socket implementation is available, there is no full compatibility w.r.t. the world of micro controller based devices which were programmed with the well known Contiki [21] operating system. Therefore, a native ARM version, of the Contiki was compiled and run as a daemon on the Linux embedded operating system. This versions has been developed by the NoES lab with NGS in order to interact with the above mentioned IEEE 802.15.4 cape for the Beagle Bone Black. Through a virtual tunneling interface on Linux, the traffic toward the 802.15.4 network is redirected to the Conitki instance (running the native-border-router-application) where the actual serialization and transmission is executed.

Summarizing, in order to provide to the BBB the access to both IoT and C-ITS networks we developed and implemented specific gateway processes for each of them. Finally, it is worth to stress that the solution that we adopted for the BTP/GeoNet is not custom but actually it fits what ETSI ITS G5 standard requires for communication gateways, notably it should be executed though interfaces in the facility layer.

4.3.3 Management and Security entities

Only basic management functionalities are included in the design, since the focus was on facility design and implementation. This applies also for the security functionalities which are not at all implemented.

Three basic service management functionalities are into the design: service registry, service activation and shutdown.

4.3.4 Facility layer

The facility layer, as seen in subsection 3.8.2, contains a set of functionalities and capabilities to support ITS applications, such as time, positioning, speed, messaging, maps. As the standard states, all Facility layer functions should be reached by other layers through some kind of Service Access Points.

Within our project, the facilities are implemented with an object oriented, component based design, where abstract classes define interfaces that expose the functionalities needed from other entities. Many concrete classes can implement the interfaces in order to open the way of different implementation but guaranteeing integration between the services and their users.

```
class GpsService {
     virtual gps_data_t get()  = 0;
}

class GpsServiceImplementation {
    gps_data_t get() {
        // actual implementation
    }
}
```

In the next sections we will show how the facility are implemented by inheriting the `ItsStation` class.

4.3.5 Applications

The Application layer contains the application that, making use of the facilities and of underneath layers, implements some sort of logic into the ITS-S. These applications should therefore have granted access to the facilities. Applications are implemented, in our project, inheriting from the class `ItsApplication`.

4.4 Implementation

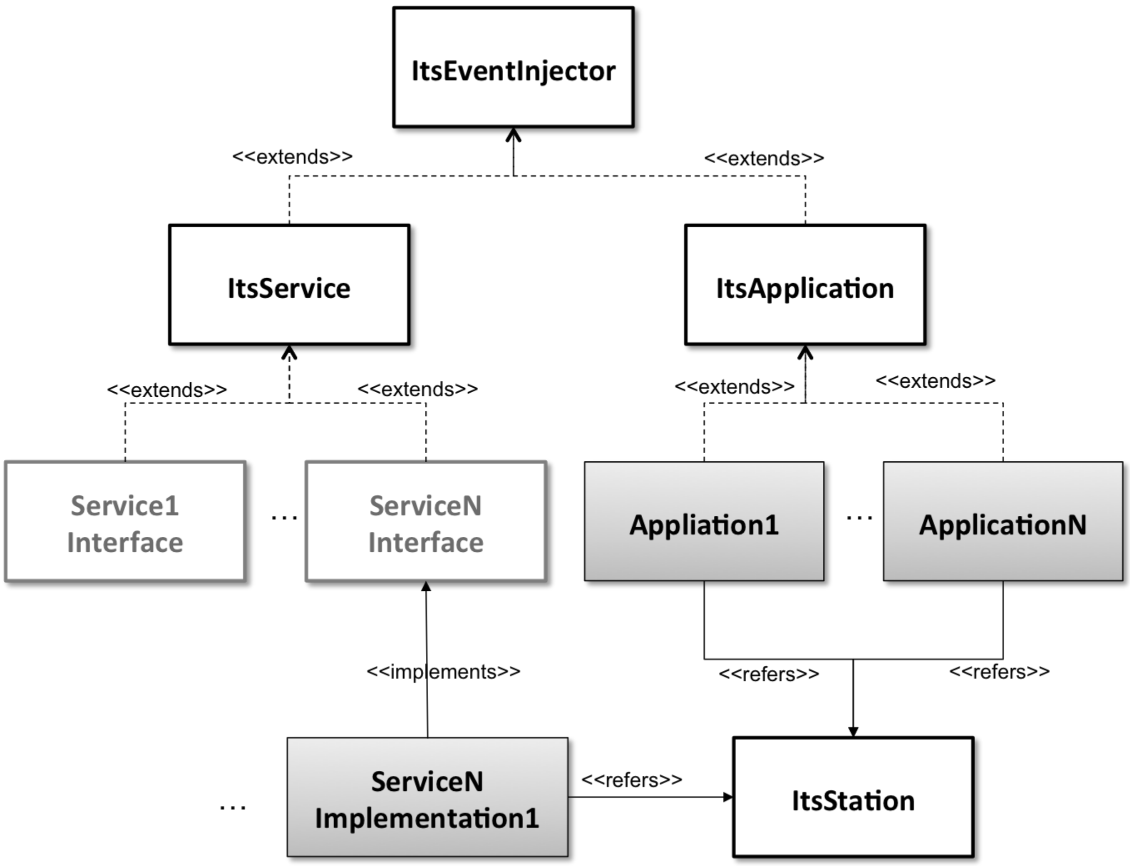

Figure 4.5: The class diagram of the framework designed for the ITS-Station.

4.4.1 Handling events

The main task of a ITS Station is handling and generating events. Events can be related to communications (discovery, connection, reception of data) or can be generated as results of internal computations, periodic events, event cause in response to other events.

Moreover, the high connection degree of the system requires to deal with many different communications (connection-based or connection-less, based on different protocols).

Therefore a core library to handle and generate requests was chosen and then wrapped into two classes defined by the framework designed to implement the Station, namely the `ItsStation` class and the `ItsEventInjector` class.

Among many libraries, we chose `libuv` [22] because of its simplicity, its small footprint and its well active community. `libuv` is the core library of the famous Nodejs, a standalone Javascript engine famous for being non-blocking, and used in many highly demanding and concurrent I/O applications.

`libuv` is the evolution of `libev`, the first implementation of I/O wrapping in Nodejs, and features a portable asynchronous and multi-thread event engine written in the C language. Furthermore the library offers a portable interface API to UDP and TCP communications, to timers and to file management.

Since `libuv` is designed around the event-driven asynchronous I/O model, it enforces an asynchronous, event-driven style of programming. Its core job is to provide an event loop and callback based notifications of I/O and other activities. It also offers core utilities like timers, non-blocking networking support, asynchronous file system access, child processes and more.

The thread pool used to offload either blocking or intensive tasks (automatically or programmatically) is set up and managed. In this way it provides a non-blocking behavior for the main thread operating on the events queue. The result is the capability to respond "almost immediately" to new events arising, especially communications requests or responses.

The library provides much more than simply abstraction over different I/O polling mechanisms: its handles and streams provide a high level abstraction for sockets and other entities. Cross-platform file I/O and threading functionality is also provided, amongst other things.

The diagram in Fig. 4.6 shows the different parts that compose `libuv` and what subsystem they relate to.

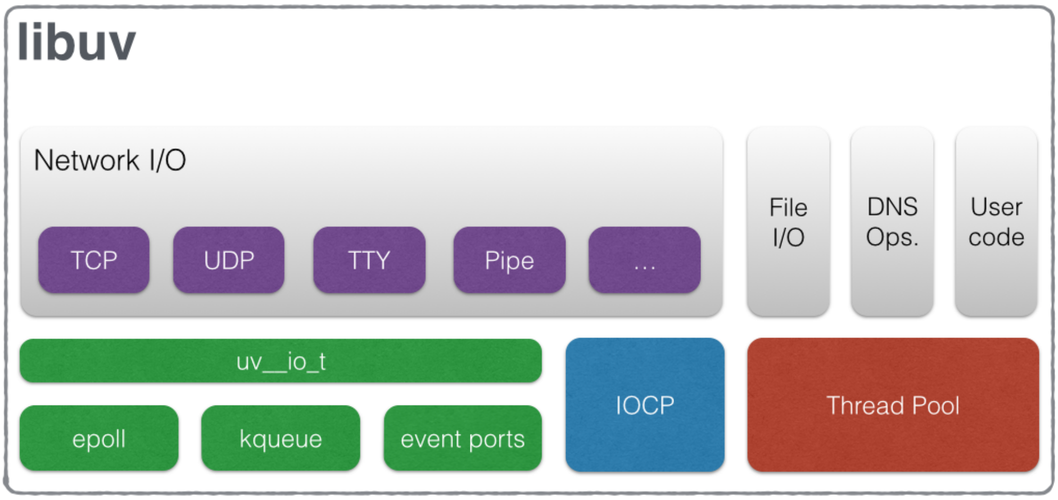

Figure 4.6: *libuv* library architecture.

4.4.2 The designed framework

The station class

The station class is the core of the designed framework. Its roles are many. It wraps and hides `libuv` with a C++11 functional interface. Callbacks passed to the libuv are hidden and the user can exploit C++11 anonymous functions to trigger actions among events. The station class also manages the service and application registries, the configuration and the shutdown of the application.

Event management

To support the creation, handling and triggering of custom events, the `ItsEventInejector` class has been designed. It defines and implements the following methods:

- `periodically` - executing a function cyclically with a given interval (wrapper against `uv_timer_start`)
- `shutdown` - receiving the shutdown state from the station and close
- `timer` - executing a function once in a given time
- `asynchronously` - enqueueing work that will be executed by a worker thread

- `on` - setting up handlers for a given event
- `trigger` - triggering a custom event within a service or application

The code to be executed is passed as a `std::Function` object, possibly created as an anonymous function that includes part of the environment where it is defined.

The Service and Application classes

The Service and Application classes abstract services (facilities, management, access) and application respectively. They both are extensions of the `ItsEventInjector` class and are therefore capable of managing events.

4.4.3 Facilities implementation

Positioning and time facility

To get the position and timing the station is provided of a GPS hardware module. The driver of the module communicates GNSS information in raw NMEA format, over a serial TTY.

To decode the raw data and access it without the need to read directly from the device, the de-facto standard `gpsd` daemon was used. `gpsd` is started from the Linux distribution at boot time, it monitors the device and exposes over a socket the collected information, in JSON format. The creators of the daemon provided an object-oriented C++ library making requests to the daemon and receiving data.

Therefore the service implemented in the `etsi-its` project, called `GpsService`, makes use of this library to collect the data. It then uses the JSON library `jsoncpp` to retrieve the needed data and to store it in a private variable.

A getter method let other facilities or applications within the station to access it in a protected way (without data races while writing the information received from the daemon).

CAN/OBDII gateway facility

On the target hardware setup, the connection toward the CAN network, on the vehicular ITS-S, is virtualized with a serial line by the OBDII connector.

Therefore direct access to the serial device is executed by the facility. Since the OBDII is able to receive many information on the status of the vehicle the CAN facility access the device periodically, collects all the supported information and stores them in private storage. This data can be accessed with a getter method by the station facilities and applications. They are, for example, accessed by the CA facility to include the information within CAMs, the moment of message generation.

The interface `CANService` defines the type of the data structure containing all the information retrieved from the vehicle internal network and the function to retrieve them. The purpose of this choice is to offer a representation of the data independent from implementations. In such a way it is the duty of the implementers to map the actual data onto the pre-defined type.

The implementation internally enables the access to the serial device and the AT protocol to access the various data.

BTP gateway facility

The BTP gateway facility enables the communication with the BTP gateway process running on the IWCUv5 board. The communication is made over Ethernet/UDP. The facility uses a shared library that defines the type of the PCI (see section 3.8.3) which is the data structure to communicate the configuration of the header for the data to be sent.

The abstract interface is defined with the class `IBtpAccessInterface` and exposes two operations:

- `virtual void btp_send(btp_options_t* opts, void* data, size_t size)`

To receive data from the GeoNet/BTP stack, an application or facility can set an handler to the "rcv" event of the service. Since the class of the service inherits from `ItsService`, it provides the `on(...)` method and the user can simply define a handler like:

```
//  f is an instance of std::function<void(void*)>
this->station->BTPAccessService->on("rcv", f);
```

4.4.4 Cooperative Awareness Facility

The Cooperative Awarness facility interface is defined in the abstract class `CAFacilityInterface`. It exposes the methods:

- `int encode(CAM_t* msg, char* buf)` which encodes the data into a buffer;
- `int decode(char *buf, size_t size, CAM_t** c)` which, starting from a buffer, decodes the data and generates the message structure;
- `int send()` which constructs a new message, encodes it and passes it to the BTP gateway interface;
- `int start_dissemination()` which activate the dissemination of CA message;
- `CAM_t* generate()` which collects all the required data from other facilities and from configuration, then generates and populates the message structure.

The `CAM_t` type is defined in the C source files generated using `asn1c` from the ASN.1 descriptor. The version used in the project is extended with an appropriate `LoadingZoneContainer` to fit the requirements of applications in logistics.

To receive a CA message, since they are propagated over the vehicular network, the facility sets a handler on the reception of BTP messages. Once a message is received, the CA facility tries to decode it: if the decoding succeeds, a new CAM has been received and this event "rcv" is thrown. Otherwise the message is not a CAM and nothing happens.

4.5 Performance evaluation

One crucial requirement is given by the CA facility functional description in the standard. The standard requires that CAM encoding is done is less than $50ms$ to support the dissemination of the messages at up to 10Hz.

On the target constrained hardware, described in section 4.2.1, we tested encoding and decoding of a 511 bytes length CAM. Our testing module generates and encodes a CAM message N times, then the bytes produced are decoded N times. The encoding phase is therefore comprehensive of the allocation and population of the `CAM_t` data structure, since the memory is re-allocated at each iteration. The resulting times from a test are given by the completion time (in microseconds, taken separately for each operation) divided N times.

This test has been repeated $N_{tests} = 1000$ times. We collected the results and calculated frequencies, mean value and standard deviation. The frequencies for the generation and encoding times are shown in Fig. 4.7 and, as the reader can see, the results are highly concentrated around the mean. The results are shown in table 4.1 and demonstrate that the requirements are largely met.

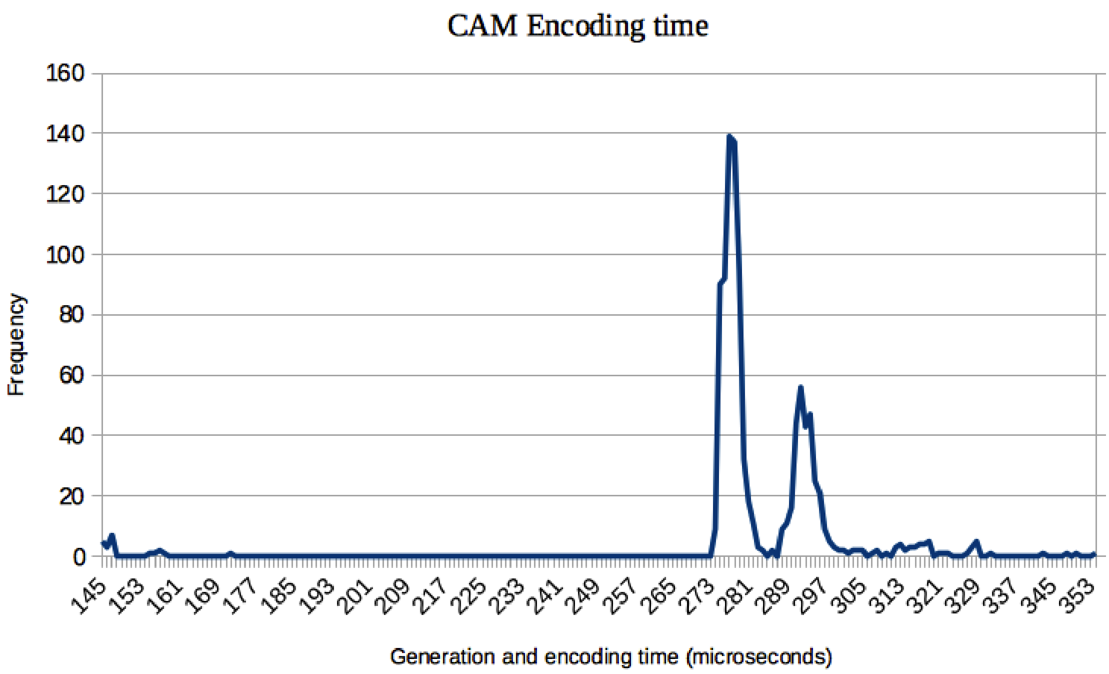

Figure 4.7: The frequencies of the completion times for generating and encoding a 511 bytes CA message. Times are in microseconds, the total number of repetitions is N_{tests}.

The results of these testing activity show that the module developed for the

CA facility meets completely the requirements set by the standards in term of encoding and decoding time of a message.

	Mean value	Standard deviation
encoding	$280\mu s$	$60\mu s$
decoding	$204\mu s$	$40\mu s$

Table 4.1: Statistics on the encoding and decoding time for CA messages over the results of M_{tests} repetitions.

Chapter 5

Developed solution in the URBELOG project

We introduced the URBELOG project in section 1.6, as a notable example of Loading/Unloading zone application. We will describe the project in more details and we will show how the developed ITS station helped to implement the parking slot sensing application.

5.1 Project goals

The URBELOG project is meant to develop and test an innovative, open, dynamic and collaborative ICT platform to support services and applications in the field of last mile logistics in urban environment. This platform will be able to aggregate the entire stakeholders ecosystem and to manage freight processes in real time from production to delivery. The project aims at realizing and validating a virtuous freight system to make goods delivery rational, cost effective and sustainable, while pushing the growth of telematics, real time B2B facilities, to be used by stockholders within cities. Pilot installations are being held in the Italian cities of Genova, Milano and Torino. These cities, represent interesting use cases of applications and will contribute to reach real world solutions that will be easily deployable all over the county.

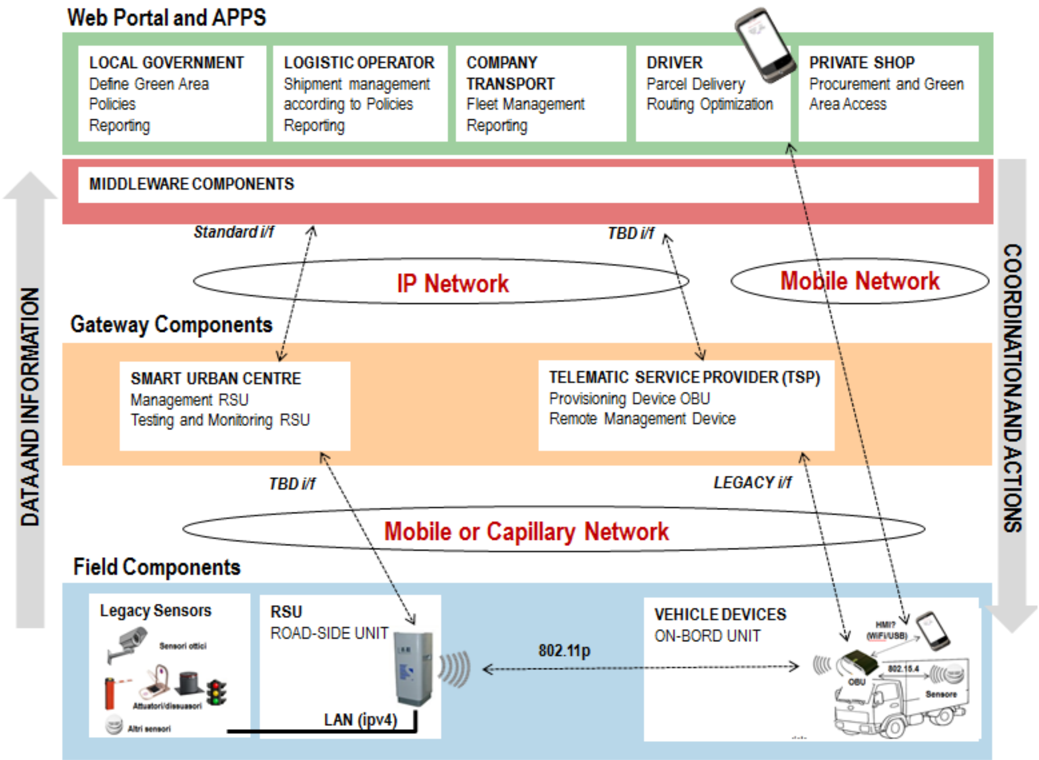

Figure 5.1: The architecture envisioned in the URBELOG project.

5.2 URBELOG architecture

Fig. 5.1 shows the architecture envisioned for the realization of the URBELOG project. Three layers are defined:

- the Web portal and APPS layer (modules to offer the applications to the final users, i.e. the logistic operators, administrative bodies, etc.),
- the Gateway components, and
- the Field components layer, where the actual devices operate.

The devices taken into account are the On Board Unit (OBU), the Road Side Unit (RSU) and the sensors.

OBU and RSU are indeed instantiations of ITS Stations and the sensors deployed on field constitute a WSN. Looking at this architecture we therefore can appreciate how much it is appropriate to test the IoT enabled ITS-Station we designed.

The duty of the RSU is to collect information from the sensors, communicate with OBUs and apply data fusion to support decisions. Both RSU and OBU are in charge of communicating constantly with the modules in the upper layers in order to have a real-time representation of the status "in the cloud", available for applications and users.

The system is designed to enable the possibility to keep data local and to execute local decisions without the need of crossing many networks to communicate. This enforces the scalability of the system, providing at the same time smaller delays for applications. The key technologies to support the locality is the usage of both IEEE 802.11p and IEEE 802.15.4 channels for the communication among devices.

5.3 Authorized parking slot application

URBELOG projects involves many applications and use cases, from freight monitoring, to parking slot discovery, navigation and reservation.

For the test site in the city of Turin, the use case to be implemented was about sensing and validating the status of a parking lot. Since the project aims to offer specific parking slots only to couriers and in particular to companies involved in the project, the parking area needs to be monitored. Notably when a vehicle stops there, the RSU needs to know if the vehicles is authorized to stop there. As a result of the design phase for this use case, six possible states of occupancy where identified for a parking slot. They are listed, with their descriptions, in Tab. 5.1.

Table 5.1: The identified possible states of a parking slot within the URBELOG project.

State name	Int Value	Description
`EMPTY`	0	There is no vehicle on the slot
`OCCUPIED`	1	A vehicle is present and not yet recognized
`ACKNOWLEDGED`	2	An authorized vehicle is present
`EXPIRED`	3	The time for authorized parking expired
`NOT_VALID`	4	The vehicle present is not authorized to stop
`UNKNOWN`	5	There is no value from the sensors

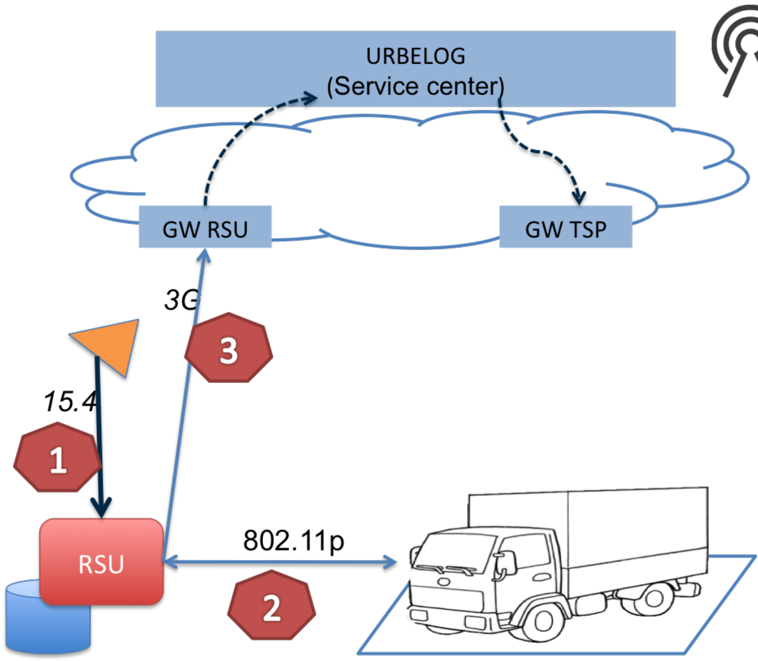

Figure 5.2: A graphical representation of the authorized parking slot application: (1) The smart camera sends to the RSU the updated status of the parking slot (occupied); (2) The RSU starts the handshake with the OBU on the vehicle to authorize the stop; (3) When the handshake is terminated the resulting new status (acknowledged or not valid) of the parking slot is communicated upstream to the URBELOG service center.

5.4 Implementation

Determining if a parking slot is occupied or not, is implemented by means of a network of wireless connected smart cameras. This devices feature computer vision algorithms to determine the occupancy of a parking slot (mapped onto a boolean value).

The authentication of a halted vehicle is instead delegated to communication between the RSU and OBU. The two devices communicate over IEEE 802.11p channel and in particular we exploited the Cooperative Awareness facility.

After the vehicle is stopped, the OBU recognizes the event (reading from the CAN/OBD II interface) and propagates an extended version of CAMs that include the information that the vehicle has halted.

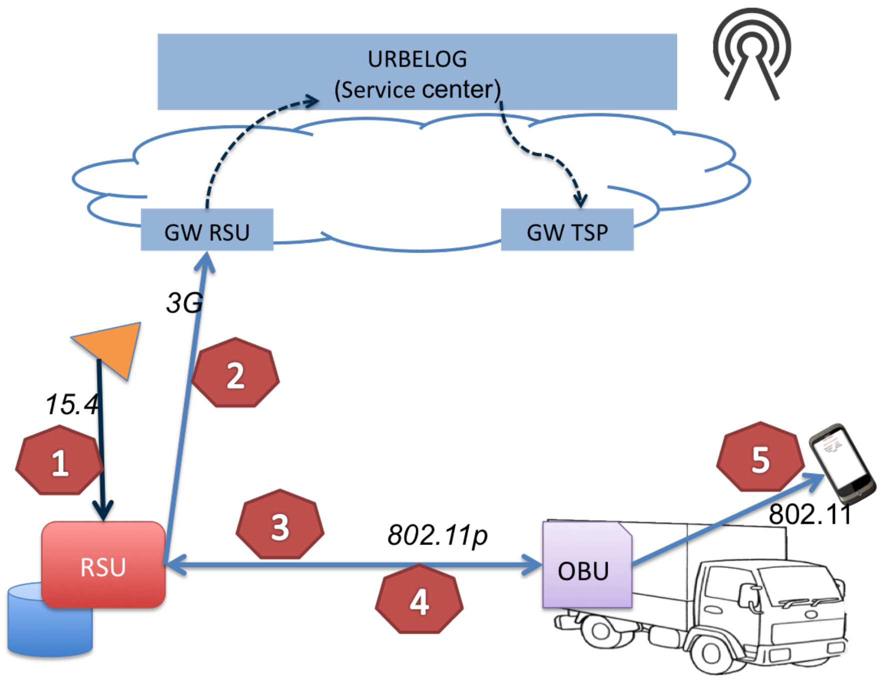

Figure 5.3: A different forseen application: when the vehicle enters the range of communication of the RSU, the available parking slots are advertised and the OBU communicates them to the driver via some HMI.

Within the RSU a hook is set: when a parked vehicle is recognized, the device checks for occupied slots and, if the vehicle is authorized, matches the two items. Every status change is communicated to the Smart Urban Center, so that action can be taken such as notification of a fine for unauthorized vehicles.

Starting from the hardware and software solution described in Chapter 4, implementing the application was straightforward: two application classes were created as descendants of the class `ItsApplication` and both the two application make use of the Cooperative Awareness facility (one constructing and sending messages, the other receiving them and to deciding if the vehicle is stopped).

5.5 On-field testing

The “smart parking slot” was successfully tested in Torino on the Nov. 13th 2015, in the facility of the Torino Telecom Italia Lab called “Open Air Lab” [23]. The smart cameras (two devices) and the RSU where installed onto street lights while the OBU was installed on a vehicle. In Fig.5.4 the devices are shown in their boxes right before being installed.

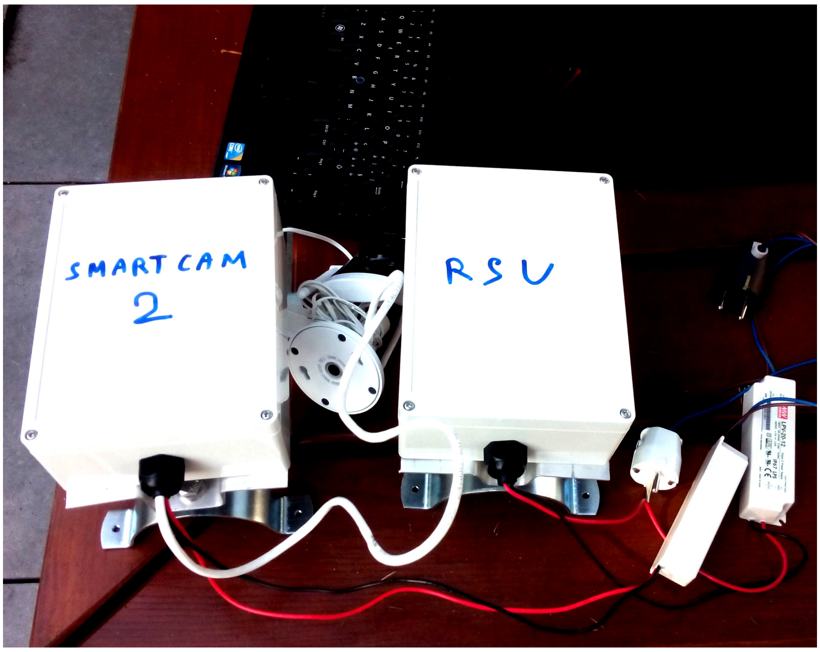

Figure 5.4: The boxes containing the prototyped devices, ready to be installed; here one smart camera and the Road Side Unit are shown.

The demo was brought on and the RSU managed to correctly understand when and which vehicle halted onto the monitored parking slot.

Chapter 6

Conclusions

In this theses, we presented our efforts and results to design and implement a sound integration between the world of automotive communication technologies and the world of the Internet of Things.

In Chapters 2 and 3, we described the paradigms, architectures and enabling technologies we were building upon. Our approach, has as a starting point the most recent evolutions in both fields and was based on normative standards. We emphasize the importance of this choice, since the wide fragmentation and the plethora of "silo" styled solutions will not be useful to achieve real developments of IoT technology in every day life. Only a vision on well funded, standard and tested integrated architectures with high interoperability capability with other systems will lead to an effective evolution and will justify investments in manufacturing and infrastructure. This investments will be indeed needed in the development of smart environments, first of all smart cities, were public administration and governmental process will be combined with market solutions.

In Chapter 4, we proposed our integrated implementation for an IoT enabled ITS Station. Beside having strong basis on the standards, our designed solution has, in every part of its architecture, the quality of proposing a novel integration involving at every layer of the communication stack. This implies a more robust and seamless integration of different communication stacks by means for example of using the CA facility to expose the status on-board and/or road-side sensors status to the other ITS-Stations.

In Chapter 5 we demonstrated the performance of our system providing a quick implementation of a use case about sensing and validating the status of some parking slots in the framework of the URBELOG project. The “smart parking slot” was successfully tested on Nov 13th 2015 in the facility of the TorinoTelecom Italia Lab, called Open Air Lab [23].

Concerning the hardware integration, the solution overcomes the lack of IEEE 802.11p modules capable of being integrated onto the main boar. Thus, exploiting the IWCUv5 board by ITRI, we provided a software interface between the main board and the IWCUv5 device, as ETSI compliant gateway facility. Moreover, the proposed prototype, integrates all the communication interfaces required by the URBELOG project, featuring the needed computing power and to leave resources to accommodate future applications.

The software has been engineered specifically to follow the concepts, guidelines and naming proposed by ETSI, by means of a component based architecture enabling isolation and easing the extensibility for further communication interfaces. It also features some non functional aspects such as responsive asynchronous behavior and easy scalability. These features are enabled by the choice of a powerful and well maintained software library suitable to abstract I/O and concurrency operations. The low level mechanisms are wrapped into high level components the instrumentation of which exploits modern features of the chosen language such as typed anonymous functions.

In conclusion, in this thesis we set a direction for easy development of meaningful applications that integrated the most innovative technologies of the Internet of Things and of the Intelligent Transport Systems. We also provide the design and the tools that we developed and that gave a robust contribution in the realization of a real world application in the field of smart cities, within the framework of the URBELOG project funded by the Italian Ministry of Education, Universities and Research and coordinated by Telecom Italia.

The proposed integration is the topic of a scientific paper by *Carignani et al.* [18] that will be presented at the 2nd IEEE World Forum on Internet of Things. The paper was one of the six selected from over 275 submitted to the conference and it will be published in the proceedings.

6.1 Future work

We consider, in the end, the future work and the possibilities opened by our prototype solution.

In the near future, a brand-new, easily extensible development board will be used to integrate the components for the ITS station and to replace the BBB, the AMBER board [24]. With the new bard, the concept of *cape* (which is the name of the extension for the BBB) becomes an *extender* of the AMBER. The new board will expose to the extensions a higher number of high speed interfaces from the System-On-Module (SOM), that will lead for example to the ability to support extenders featuring 4G/LTE communication interfaces.

The new version based on the AMBER board will also feature a customizable System On Module. The SOM supported at the moment are produced by Variscite and feature Freescale MX6 processor (with 1, 2 or 4 cores). This means that this kind of platforms will soon (or already do) feature many-core processing units, prevalently ARM models. The important implications of many-core general purpose processing architectures in embedded devices are studied in *Yamauchi et. al.* [25]: multi core architecture will be the major CPU technology that will meet real time multi source technology computations and power saving efficiency.

Regarding the software architecture of the ITS Station, encapsulation and isolation of services and applications will be fully implemented. The management features set by the ETSI standards will be provided at the high level to enable installation, activation, deactivation and discover of packages or modules but also to handle dynamic re-configuration, customization and remote control. An important aspect that will be evaluated is the runtime support to provide the management features. Among the nowadays proposed solutions, virtualization technologies are of relevant potential.

Beside that, many other specific application could be designed in the broad field of IoT for Intelligent Transport Systems. Finally the promotion of standardization initiatives at ETSI OneM2M and ITS working groups will be pursued.

Bibliography

[1] G. Pellerano, M. Falcitelli, M. Petracca, and P. Pagano, “6lowpan conform its-station for non safety-critical services and applications,” in *ITS Telecommunications (ITST), 2013 13th International Conference on*, Nov 2013, pp. 426–432.

[2] “ETSI TR 102 638 V1.1.1 Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Definitions,” http://www.etsi.org/deliver/etsi_tr/102600_102699/102638/01.01.01_60/tr_102638v010101p.pdf, accessed: 2015-09-01.

[3] “Mckinsey global institutedisruptive technologies:advances that willtransform life, business,and the global economy,” http://www.slideshare.net/brandsynapse/mgi-disruptive-technologiesfullreportmay2013.

[4] E. Borgia, “The internet of things vision: Key features, applications and open issues,” *Computer Communications*, vol. 54, pp. 1–31, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2014.09.008

[5] F. Pacini, “Performance of data serialization methods for wireless communication in resource-constrained devices,” 2014.

[6] “Tizen: An open source, standards-based software platform for multiple device categories,” https://www.tizen.org/.

[7] A. Festag, “Cooperative intelligent transport systems standards in europe,” *Communications Magazine, IEEE*, vol. 52, no. 12, pp. 166–172, 2014.

[8] “802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular

Environments”,” IEEE 802.11p published standard, Institute of Electrical and Electronics Engineers, July 2010.

[9] “ETSI EN 302 663 V1.2.0 IIntelligent Transport Systems (ITS); Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band ,” http://www.etsi.org/deliver/etsi_en/302600_302699/302663/01.02.00_20/en_302663v010200a.pdf, accessed: November 2015.

[10] “ETSI TS 102 637-1 V1.1.1 Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 1: Functional Requirements,” http://www.etsi.org/deliver/etsi_ts/102600_102699/10263701/01.01.01_60/ts_10263701v010101p.pdf, accessed: November 2015.

[11] E. Belyaev, A. Vinel, M. Jonsson, and K. Sjoberg, “Live video streaming in ieee 802.11p vehicular networks: Demonstration of an automotive surveillance application,” in *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on*, April 2014, pp. 131–132.

[12] “ITSSv6 project,” http://itssv6.eu, accessed: 2015-09-01.

[13] L. Virág, J. Kovács and A. Edelmayer, “Extension of the its station architecture to low-power pervasive sensor networks,” in *Proceedings of the IEEE 27th International Conference on Advanced Information Networking and Applications Workshops (WAINA2013)*, Barcelona, Catalonia, Spain, Mar. 2013, pp. 1386–1391.

[14] “FOTsis project,” http://www.fotsis.com/, accessed: 2015-09-01.

[15] “ICSI project,” http://www.ict-icsi.eu, accessed: 2015-09-01.

[16] A. Azzara’, M. Petracca, and P. Pagano, “The icsi m2m middleware for iot-based intelligent transportation systems,” in *International Workshop on cooperative sensing for smart mobility (COSSMO) Co-located with Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*, Sep 2015.

[17] "SAE Standard J1962, Diagnostic Connector," http://standards.sae.org/j1962_201509/, accessed: 2015-09-01.

[18] M. Carignani, S. Ferrini, M. Petracca, M. Falcitelli, and P. Pagano, "A prototype bridge between automotive and the iot," in *Internet of Things (WF-IoT), 2015 IEEE World Forum on*, December 2015.

[19] "New generation sensors," http://ngs-sensors.it/.

[20] "Microchip MRF24J40 Datasheet," http://www.microchip.com/downloads/en/DeviceDoc/39776C.pdf, accessed: 2015-09-01.

[21] "Contiki: The open source OS for the internet of things," http://www.contiki-os.org/, accessed: 2015-09-01.

[22] "libuv, a multi-platform support library with a focus on asynchronous i/o." http://docs.libuv.org/.

[23] "Tilab, telecom italia open air lab," http://www.telecomitalia.com/tit/en/innovazione/i-luoghi-della-ricerca/tilab.html.

[24] "Amber open platform designed, general purpose gateway platform iot." http://www.amber-lab.com/.

[25] T. Yamauchi, H. Kondo, and K. Nii, "Automotive low power technology for iot society," in *VLSI Technology (VLSI Technology), 2015 Symposium on*. IEEE, 2015, pp. T80–T81.

Acknowledgements

This work, which marks the conclusion of this Master Degree, would not have been possible without all the people, and they are many, who taught, helped, inspired and supported me in the last years. Even without the ability to express it appropriately, I feel the importance to write some lines here.

I would like to thank Dr. Paolo Pagano for the opportunity to study in the Network of Embedded Systems lab and to work there for my thesis under the supervision of Prof. Piero Castoldi. Thanks to all the people who worked with me in the lab, namely Mariano, Matteo, Claudio, Stefano, Kibrom and Silvia, for their support but mostly for every smile and laugh they shared everyday in the last months.

Thanks to all the professors of the Master Degree for the dedication and passion they put to offer a high quality learning experience.

Thanks to my colleagues at the Computer Science department, the one that I met in my first years of University, the ones I met recently, the ones that chose different paths and the one who shared this journey till the very last day. Thanks to all of you, because you were the first to understand me and we shared some really unforgettable moments.

I really want to thank all my friends from my city, Lucca. I should especially thank the ones (and they are many) who shared long rides back and forth to Pisa. But mostly thanks to all the people that supported me, shared good and bad moments, listened to me. I feel lucky to have you beside me.

I'm grateful to and for my beautiful family (all of their names would not fit in a page) who always made me feel loved. Thanks to Giulia, Elisa and Uberto who are the best in understanding me and always saying the right thing. Finally, thanks to my parents Ilaria and Gino for giving me this opportunity, fostering me to do my best while always offering their trust and inspiration.