UNIVERSITÀ DEGLI STUDI DI BRESCIA
FACOLTÀ DI INGEGNERIA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

DOTTORATO DI RICERCA IN TECHNOLOGY FOR HEALTH

CICLO XXXIII

TESI DI DOTTORATO DI RICERCA

New LoRaWAN solutions and cloud-based systems for eHealth applications

Ing-inf/07

Dhiego Fernandes Carvalho

Relatore: **Prof.ssa Alessandra Flammini**

Correlatore: **Prof. Paolo Ferrari**

Coordinatore del Dottorato

Prof.ssa Alessandra Flammini

UNIVERSITÀ
DEGLI STUDI
DI BRESCIA

DOTTORATO DI RICERCA IN TECHNOLOGY FOR HEALTH

Ing-Inf/07

CICLO XXXIII

New LoRaWAN solutions and cloud-based systems for eHealth applications

Dhiego Fernandes Carvalho

Firma .....

Relatore: **Prof.ssa Alessandra Flammini**

Firma .....

Correlatore: **Prof. Paolo Ferrari**

Firma .....

Coordinatore del Dottorato: **Prof.ssa Alessandra Flammini**

Firma .....

Acknowledgments

I would like to thank my PhD supervisor, Alessandra Flammini, for being part of this great research group in Technology for Health.

My special gratitude to Emiliano Sisinni for helping me with your patience and great technical skills in my work and Paolo Ferrari, including his family, for not being just a professor and tutor, but a friend who also supported me a lot during my stay in Italy.

My colleagues in Open Space for the great times lunching and taking coffee together during the endless hours studding sharing the same workplace.

Obviously, I would not be here if were not my parents and family who gave me the financial support in my lifetime and showed me how to accomplishment my goals. Not forgetting, and most important, my daughter, Diana, who is the main reason why I am abroad doing my PhD and gives me strength to move forward.

Contents

Abstract

This PhD thesis belongs to the area of the Technology for Health, a field that is rapidly expanding beyond the pure medical applications. Frequently, new technologies coming from diverse sources are incorporated into Health systems in order to increase the overall benefit.

In particular, the Internet of Things (IoT) promises to deeply change the way we live and work thanks to the availability of ubiquitous smart devices, as in case of recent examples of Smart City. However, the practical integration of these devices with cloud applications is sometimes a complex and hard task, especially because of the complexity of their physical connectivity. For instance, in case of wireless, the specific requests of each application scenario deeply affect the usability and the effectiveness. In this context, medical devices require more flexibility and connectivity in order to mitigate human's errors, avoid medical accidents, improve medical response in emergency cases and anticipate diagnosis. Considering such situation, this thesis aims to investigate new IoT solutions and cloud-based system for eHealth applications. In order to clarify how this work is organized, the **Figure 0-1** shows the thesis workflow, highlighting the main steps and contributions.

The first step of this work was a broad investigation about potential of IoT in medical and diagnostic applications. A Medical cyber-physical systems (MCPS) was suggested as a possible approach for detecting and limiting the impact of errors and wrong procedures for Neurodegenerative Disease Diagnosis. The proposed MCPS leverages on atypical cloud-based architecture: it uses a message-oriented middleware, complemented by historical database for further data processing. AMQP and MQTT protocols, two of the most diffused message oriented protocols for cloud data exchange, were evaluated. The focus of experimental setup is the real-time performance, which has the most challenging requirements. Time-related metrics spread across MCPS confirm the feasibility of the proposed architecture for real-time diagnosis and collaboration among researchers, since the end-to-end delay is on the order of few tens of milliseconds for local networks and up to few hundreds of milliseconds for wide range networks. This research has paved a solid baseline in Health Systems for further researches in the IoT scenario.

The experimental validation of the considered MCPS demonstrates the effectiveness of the IoT paradigm applied to the health scenario, but also permitted to highlight the limitations related to wired connectivity, which greatly limits the overall system flexibility, especially

from the patient point of view. Wireless technologies may solve such problems, and this is the reasons why during this work they came into attention and were investigated.

Since the beginning it was clear that not all the wireless systems are suitable for satisfying all the requirements of MCPS. Thus, it was initiated the investigation of the state of art of wireless connectivity specific for Wide Area Network. The coverage of large area is very important for medical systems and diagnostic applications, since allow seamless applications inside and outside buildings, expanding the usability of MCPS outside the hospital.

As a result, the thesis research focus moved to the Low Power Wide Area Network technologies, which is the newest and most promising solutions in this specific field. LPWAN networks emerged in the recent past as possible answer to IoT demands since they better fit the large coverage, low-cost, low-power consumption and simple infrastructure management of legacy wireless solutions. Even a plenty of LPWAN technologies have appeared, two solutions emerged and currently have significant commercial installations and operators around the world, i.e. SigFox and LoRaWAN. The latter is an open standard relying on a proprietary radio whose final goal is an alternative approach to mobile networks. The former is a proprietary and patented solution that exploits ultra narrow band modulation in order to transmit up to 140 messages per day per node, with a maximum length of 12B. However, both were created for Internet of Thing applications into of smart cities context, speacilly for smart metering applications that demand few transmissions from a huge number of end nodes. Therefore, many companies and governments have started to encourage the adoption of these two main technologies in the IoT scenario spread in the globe. Several SigFox carriers started their activities and already provide services within the IoT ecosystems [1.50]. In the other side, LoRaWAN is maintined by the LoRa Alliance [1.51], which is a group formed by developers, manufacturers and operators.

Inside the LPWAN scope, it is straightforward the choice of LoRaWAN technology for further researches in this work. LoRaWAN is a great solution, because it is based on an open stardand maintained by a global alliance worldwide. Also, LoRaWAN has a good performance in static conditions and it is fully supported by many delopment kits and hardware modules, in this way facilitating the implementation of a real world experimental setup. After an investigation in the available literature, it has been found that LoRaWAN can be used for monitoring human body activity, e.g.: in case if the person is stopped or running, [1.42]. Also the monitoring of Urinary Track Infection [1.43], Heart Failure [1.44], Blood Pressure are possible. LoRaWAN is used to transmit Glucose, Temperature and others vital signs [1.45]-[1.47]. Last, in the middle of Coronavirus Pandemic, there are plenty of LoRaWAN

applications available on LoRa Alliance website for helping fight Covid-19 [1.48]. As a consequence, LoRaWAN technology demonstrate itself as the most promising solution (especially, in smart city context) and the work has been oriented to integrate it with eHealth applications.

A test methodology was evaluated for the experimental assessment of architectural delays of public and private LoRaWAN networks. The use and the effectiveness of the proposed methodology are shown by means of use cases that involve different LoRaWAN implementations and several hardware platforms. The results showed that test methodology can be used to compare the performance of entire LoRaWAN systems, helping the owner of the infrastructure and the user in making choices and optimizations.

Continuing to stress LoRaWAN technology, it was mixed a LoRaWAN modem and localization techniques, as the Global Positioning Systems (GPS, outdoor) and real-time location systems (RTLS, indoor), for mixed (indoor and outdoor) scenarios (e.g. a Smart Campus applications). Moreover, a LoRaWAN node embedded with both GPS and Ultra Wide Band based UWB-RTLS was developed and tested in real-world scenarios. The result, again, shows the feasibility of the proposed approach, demonstrating that localization errors are in the order of few tens of meters and few of tens centimeters for GPS and UWB, respectively.

Furthermore, the LoRaWAN technology can be used on mobile systems, as demonstrated by a short research in the automotive field. LoRaWAN has been used as a backup network for Industry 4.0 enabled cars. During this research, a smart Edge device is plugged to an intelligent vehicle for checking its internal functions when the car is on the road. Usually, the Edge device has 3G/4G modems, but in this case it was extended by means of an additional LoRaWAN modem for backup situation (in case of lacking of 3G/4G connectivity). This research aims to demonstrate the application constrains and design directions to achieve a correct integration between LoRaWAN infrastructure and existing framework and concepts (in this case it was integrated with the Internet of Intelligent Vehicles for Industry 4.0).

The previous studies confirmed that the LoRAWAN technology is a good and solid start point for MCPS. However, there were important point missing: coverage range and transmission delivery rate. Even LoRaWAN is famous for km-long ranges, the coverage is (in many situations) still too small for safe operation in medical/emergency applications. Also delivery ratio of data packet is often below 70%, again too small for critical applications.

For this reasons, the next part of this work is devoted to enhance LoRaWAN architecture by means of:

- Extending LoRaWAN Coverage: developing an extender node based on commercially available hardware, named as e-Node, to expand its coverage into IoT scenario. This proof-of-concept prototype consists in replicating LoRaWAN messages being transparent to the network. The measurements campaigns, obtained with a purposely designed test bench, showed the feasibility of this approach with the doubling of the coverage range.
- Improving LoRaWAN feasibility in Emergency situations: using the same device from the LoRa Extender, the e-Node, a LoRa-REP access method was proposed to enhance LoRaWAN technology in emergency situations. The measurements campaigns, using fixed and mobile nodes in public and private LoRaWAN networks, demonstrate this mechanism increases the LoRaWAN performance, increasing delivery ratio up to 99%.

The researches conducted to investigate, validate and enhance LoRaWAN architecture demonstrated this promise technology is an affective answer to the request of wireless connectivity over large areas, when communications latencies comparable with human activity is tolerable.

In the last part of this work, the LoRaWAN technology with all the previously introduced enhancement has been applied to elderly care applications. In this research, the LoRaWAN technology is evaluated for possible complementing indoor and outdoor positioning solutions, in order to alert a remotely-connected caregiver about accidents (e.g., falls). The use of smart, low-cost peripheral modules speeded up the development of a proof-of-concept prototype, able to demonstrate the feasibility of the proposed approach in terms of communication latency and (indoor) localization error. The final result results demonstrated sub-meter error in a typical indoor scenario, with average communication latency below 1 s, and a high delivery ratio.

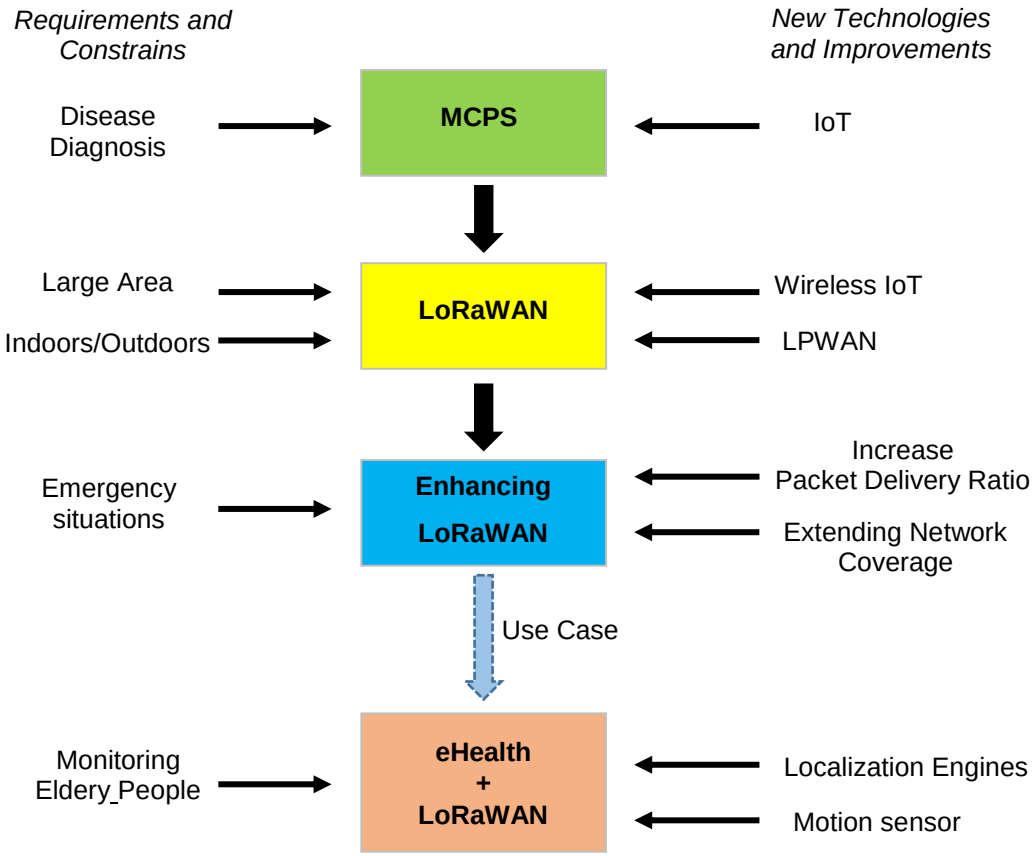


***Figure 0-1**: Graphical overview of the Thesis workflow*

Sintesi

Questa tesi di dottorato appartiene all'area della Tecnologie per la Salute, un campo in rapida espansione che va oltre le pure applicazioni mediche. Spesso, nuove tecnologie provenienti da diverse campi applicativi vengono incorporate nei sistemi sanitari al fine di aumentare i benefici ottenibili.

In particolare, l'Internet of Things (IoT) promette di cambiare profondamente il modo in cui viviamo e lavoriamo grazie alla pervasività di dispositivi smart, come nel caso dei recenti esempi di Smart City. Tuttavia, l'integrazione pratica di questi dispositivi con le applicazioni cloud è a volte un compito difficile, soprattutto a causa delle limitazioni dettate dalle soluzioni di connettività. Ad esempio, nel caso del wireless, le richieste specifiche di ogni scenario applicativo incidono profondamente sull'usabilità e sull'efficacia. Nel caso dei dispositivi medici, questi richiedono maggiore flessibilità e affidabilità al fine di mitigare gli errori umani, evitare incidenti medici, migliorare la risposta medica in casi di emergenza e anticipare le diagnosi. Considerando tale situazione, questa tesi mira a indagare nuove soluzioni IoT e sistemi basati su cloud per applicazioni di eHealth. Al fine di chiarire come è organizzato questo lavoro, la Figure 0-1 mostra il flusso di lavoro della tesi, evidenziandone i principali passaggi e contributi.

Il primo passo di questo lavoro è stata un'ampia indagine sul potenziale dell'IoT nelle applicazioni mediche e diagnostiche. È stato suggerito un sistema medico cyber-fisico (MCPS - Medical cyber-physical systems) come possibile approccio per rilevare e limitare l'impatto di errori e procedure errate nella diagnosi delle malattie neurodegenerative. L'MCPS proposto fa leva su un'architettura innovativa basata su cloud: utilizza un middleware orientato ai messaggi, integrato da un database storico per un'ulteriore elaborazione dei dati. Sono stati valutati i protocolli AMQP e MQTT, due dei più diffusi protocolli orientati ai messaggi per lo scambio di dati nel cloud. L'obiettivo dell'analisi, che si riflette nella configurazione sperimentale, è il rispetto di requisiti (stringenti) di tempo reale del sistema. Le metriche relative alle tempistiche sistema MCPS creato confermano la fattibilità dell'architettura proposta per la diagnosi e la collaborazione in tempo reale tra i ricercatori in campo medico: infatti il ritardo end-to-end è dell'ordine di poche decine di millisecondi per le reti locali e di poche centinaia di millisecondi per reti geografiche. Questa ricerca ha costituito una solida base per ulteriori ricerche sul paradigma IoT applicato allo scenario dei sistemi sanitari.

Se da un lato la validazione sperimentale dell'MCPS considerato ne ha dimostrato l'efficacia, dall'altro ha anche permesso di evidenziare alcuni limiti legati alla connettività

cablata; quest’ultima infatti limita notevolmente la flessibilità complessiva del sistema, soprattutto dal punto di vista del paziente. Le tecnologie wireless possono risolvere tali problemi, ed è questo il motivo per cui durante questo lavoro di tesi si è deciso di spostare l'attenzione verso il wireless avviando un nuovo filone di ricerca.

Fin dall'inizio era chiaro che non tutti i sistemi wireless sono adatti a soddisfare tutti i requisiti di un MCPS. È stata così avviata l'indagine sullo stato dell'arte della connettività wireless nel caso specifico delle grandi aree di coperture, quindi delle Wide Area Network. La copertura di una vasta area è molto importante per i sistemi medici e le applicazioni diagnostiche, poiché consente applicazioni senza soluzione di continuità all'interno e all'esterno degli edifici, ampliando l'usabilità dell'MCPS anche al di fuori dell'ospedale.

Di conseguenza, il focus della ricerca della tesi si è spostato sulla tecnologia Low Power Wide Area Network, che è la soluzione più recente e più promettente in questo campo specifico. Le reti LPWAN sono emerse nel recente passato come possibile risposta alle richieste dell'IoT, poiché si adattano meglio all'ampia copertura, al basso costo, a basso consumo energetico e alla semplice gestione dell'infrastruttura delle soluzioni wireless legacy. In verità sono state create molte tecnologie LPWAN, tra le quali sono emerse due soluzioni chee attualmente hanno importanti installazioni commerciali e operatori in tutto il mondo: SigFox e LoRaWAN. Quest'ultimo è uno standard aperto che si basa su una radio proprietaria il cui obiettivo finale è un approccio alternativo alle reti mobili. La prima invece è una soluzione proprietaria e brevettata che sfrutta la modulazione a banda ultra stretta per trasmettere fino a 140 messaggi al giorno per nodo, con una lunghezza massima di 12B. Tuttavia, entrambi sono stati creati per applicazioni IoT nel contesto delle "smart cities", in particolare per applicazioni di misurazione intelligente, che richiedono poche trasmissioni da un numero enorme di nodi finali. Pertanto, molte aziende e governi hanno iniziato a incoraggiare l'adozione di queste due principali tecnologie nello scenario IoT su scala globale. Diversi provider SigFox hanno iniziato le loro attività e forniscono già servizi all'interno degli ecosistemi IoT [1.50]. Dall'altro lato, LoRaWAN è gestito dalla LoRa Alliance [1.51], che è un gruppo formato da sviluppatori, produttori e operatori.

All'interno dell'ambito LPWAN, sembra più coerente con questo lavoro la scelta della tecnologia LoRaWAN. LoRaWAN è un'ottima soluzione per studiare ulteriori espansioni, perché si basa su uno standard aperto. Inoltre, LoRaWAN ha buone prestazioni in condizioni statiche ed è completamente supportato da molti development kit e moduli hardware, facilitando in questo modo l'implementazione di setup sperimentali nel mondo reale.

Dopo una indagine nella letteratura scientifica disponibile, è stato ampiamente chiaro che LoRaWAN può essere usato per monitorare l'attività del corpo umano. Ad esempio è stato utilizzato per capire se la persona è ferma o in movimento, [1.43], se ha infezione delle vie urinarie [1.43], se soffre di insufficienza cardiaca [1.44], e per monitorare pressione sanguigna, glucosio, temperatura e altri segni vitali [1.45]-[1.47]. Nel mezzo della pandemia di Coronavirus, sono state mostrate molte applicazioni LoRaWAN disponibili per aiutare a combattere il Covid-19, come visibile sul sito web di LoRa Alliance [1.48]. Di conseguenza, la tecnologia LoRaWAN si è dimostrata la soluzione più promettente (soprattutto, nel contesto delle smart city) e il lavoro è stato orientato ad integrarla con le applicazioni eHealth.

È stata valutata una metodologia di test per la valutazione sperimentale dei ritardi architetturali delle reti LoRaWAN pubbliche e private. L'utilizzo e l'efficacia della metodologia proposta sono dimostrati attraverso casi d'uso che coinvolgono diverse implementazioni LoRaWAN e diverse piattaforme hardware. I risultati hanno mostrato che la metodologia di test può essere utilizzata per confrontare le prestazioni di interi sistemi LoRaWAN, aiutando il proprietario dell'infrastruttura e l'utente nelle scelte e nelle ottimizzazioni.

Successivamente è stata applicata la tecnologia LoRaWAN ad un sistema reale, unendo un modem LoRaWAN con moduli di localizzazione, e utilizzando soluzioni come i sistemi di posizionamento globale (GPS, outdoor) e i sistemi di localizzazione in tempo reale (RTLS, indoor), con l’obiettivo di utilizzo in scenari misti indoor e outdoor (es. applicazioni Smart Campus). Nello specifico, è stato sviluppato un nodo LoRaWAN integrato con GPS e UWB-RTLS basato su banda ultra larga e testato in scenari del mondo reale. Il risultato, ancora una volta, mostra la fattibilità dell'approccio proposto, dimostrando che gli errori di localizzazione sono dell'ordine di alcuni metri per il GPS e di poche decine di centimetri per UWB.

Inoltre, la tecnologia LoRaWAN può essere utilizzata su sistemi mobili, come dimostrato da una ricerca fatta in campo automobilistico. In questo caso LoRaWAN è stato utilizzato come rete di backup per le auto abilitate per l'Industria 4.0. Durante questa ricerca viene collegato un dispositivo smart Edge a un veicolo intelligente per verificarne le funzioni interne quando l'auto è in viaggio. Solitamente il dispositivo Edge dispone di modem 3G / 4G, ma in questo caso è stato ampliato tramite un modem LoRaWAN aggiuntivo per situazioni di backup (in caso di mancanza di connettività 3G / 4G). Questa ricerca mira a dimostrare i vincoli applicativi e le direzioni di progettazione per ottenere una corretta integrazione tra l'infrastruttura LoRaWAN e il framework ei concetti esistenti (in questo caso è stato integrato con Internet of Intelligent Vehicles for Industry 4.0).

Gli studi precedenti hanno confermato che la tecnologia LoRAWAN è un buon e solido punto di partenza per realizzare un MCPS. Tuttavia, mancavano due punti importanti: il raggio di copertura è ancora limitato e il numero di pacchetti trasmessi che viene ricevuto correttamente è basso. Anche se LoRaWAN è famoso per i chilometri di raggio di copertura, il range è (in molte situazioni) ancora troppo piccolo per un funzionamento sicuro in applicazioni mediche / di emergenza. Anche il rapporto di consegna del pacchetti ricevuti/pacchetti trasmessi è spesso inferiore al 70%, ancora una volta troppo piccolo per applicazioni critiche.

Per questi motivi, la parte successiva di questo lavoro di tesi è stata dedicata al miglioramento dell'architettura LoRaWAN mediante:

- Estensione della copertura LoRaWAN: sviluppo di un nodo extender basato su hardware disponibile in commercio, denominato e-Node, per espandere la copertura nello scenario IoT. Questo prototipo proof-of-concept consiste nel replicare i messaggi LoRaWAN in modo trasparente per la rete. Le campagne di misura, ottenute con un banco prova appositamente progettato, hanno evidenziato la fattibilità di questo approccio che fornisce il raddoppio del range di copertura.
- Miglioramento dell'usabilità di LoRaWAN in situazioni di emergenza: impiegando lo stesso dispositivo dal LoRa Extender, l'e-Node, è stato proposto un metodo di accesso, chiamato LoRa-REP, per migliorare la tecnologia LoRaWAN in situazioni di emergenza. Le campagne di misurazione (utilizzando nodi fissi e mobili in reti LoRaWAN pubbliche e private) hanno dimostrato che questo meccanismo accresce le prestazioni di LoRaWAN, aumentando il rapporto di consegna di pacchetti corretti fino al 99%.

Le ricerche fatte per studiare, convalidare e migliorare l'architettura LoRaWAN hanno dimostrato che questa tecnologia è una risposta efficace alla richiesta di connettività wireless su vaste aree, quando sono tollerabili latenze di comunicazione comparabili con i tempi di reazione dell'attività umana.

Nell'ultima parte di questo lavoro, la tecnologia LoRaWAN con tutti i miglioramenti precedentemente introdotti è stata infine applicata ad un caso d'uso relativo all'assistenza agli anziani. In questa ricerca, la tecnologia LoRaWAN viene valutata per possibili soluzioni di posizionamento interno ed esterno complementari, al fine di avvisare un infermiere connesso a distanza in caso di incidenti (ad esempio, cadute). L'utilizzo di moduli periferici intelligenti e a basso costo ha accelerato lo sviluppo di un prototipo proof-of-concept, in grado di

dimostrare la fattibilità dell'approccio proposto in termini di latenza di comunicazione ed errore di localizzazione (indoor). I risultati finali hanno dimostrato un errore inferiore al metro in un tipico scenario indoor, e una latenza di comunicazione media inferiore as 1 s e un elevato rapporto di pacchetti corretti consegnati.

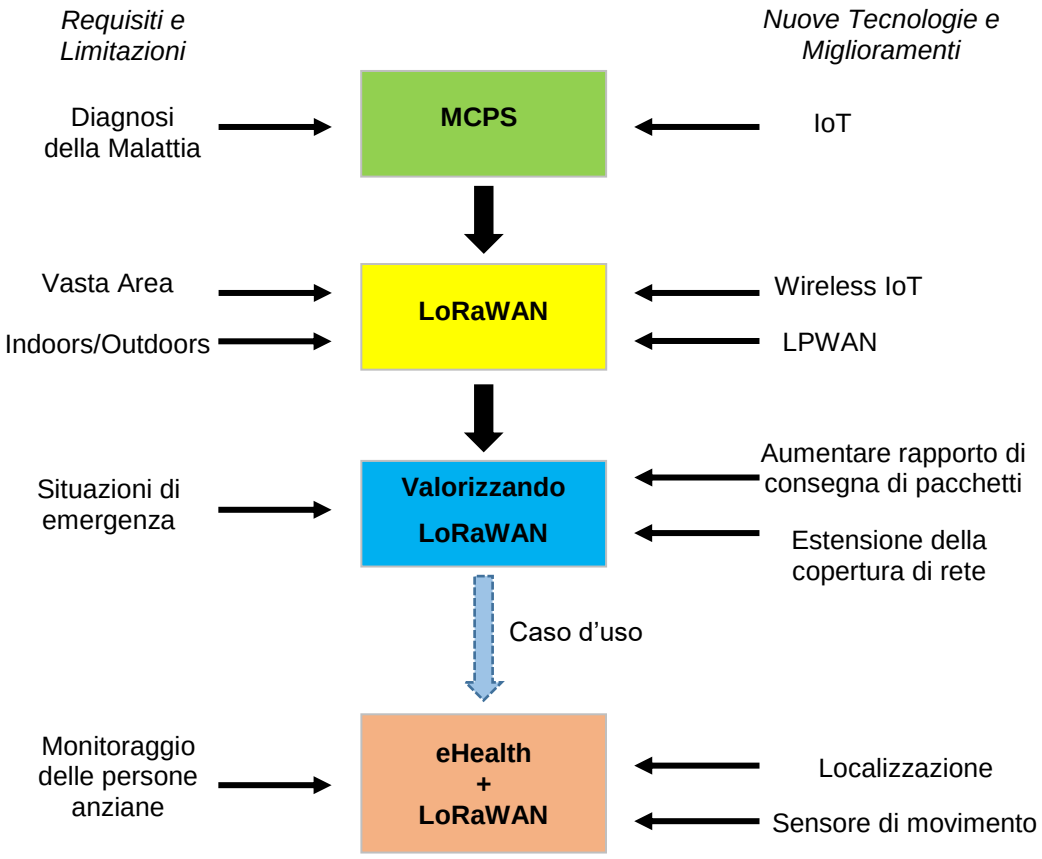


***Figure 0-1**: Panoramica grafica del flusso di lavoro della tesi*

Introduction

Internet of Things (IoT) is a trend in this technological world. Devices being connected everywhere, every time and in any media is an outstanding transformation in how people interacts between them and how they use technology. The letter "I" of IoT means that Internet is the central player in the communication, but how data is manipulated and transmitted toward the network depend on the demands of each application.

In the light of these considerations, several researches are ongoing for creating eHealth applications, especially for elderly people. One example is the ambient assisted living (AAL) technologies allowing the elderly to easily look for help in emergency situations. AAL uses the Internet of Things' network infrastructure, which enable medical devices receive and transmit data to the cloud. In this way, Low Power Wide Area Networks (LPWANs) helps to answer the demands of wireless connectivity over large areas, when data transmission latencies comparable with daily human activity is acceptable. The LoRaWAN technology has emerged as a promising and versatile solution inside LPWAN scope and it has been currently largely adopted in many Smart Cities scenarios e.g., for smart metering, since it allows the possibility to use both public and private communication networks.

During these three PhD academic years, a plenty of researches were evaluated in order to investigate technologies to be applied in eHealth applications inside IoT scenario. Most parts of these works are inside LPWAN area, especially in LoRaWAN technology. Furthermore, this thesis synthesizes the main outcomes produced in the PhD program, divided in four phases:

- **First Phase**: identification of suitable application scenarios, mainly focusing on health contexts.
- **Second Phase**: after a deep investigation on available LPWAN technologies, evaluate the most suitable one (LoRaWAN) to be applied with health devices. In order to validate this promising technology for data communication, storage and processing. Resulting in development and implementation of methodologies and tools for evaluating LoRaWAN network performance;
- **Third Phase**: Development and implementation of proof-of-concept for enhancing the performance and feasibility of LoRaWAN infrastructure.
- **Forth Phase**: evaluation of the LoRaWAN technology integrated with health applications for specific on-field testing in an Internet of Things scenario.

This thesis is organized according to these four phases mentioned, resulting in four chapters as Follow:

- **Chapter 1**: as the part of initial research in health area, this chapter aims to investigate and address an IoT Communication Architecture. Using MQTT and AMQP protocols for cloud data exchange in Medical Cyber-Physical System (MCPS) applied to a sample use case (Neurodegenerative Disease Diagnosis). This MCPS has paved a solid baseline in Health System area for further researches in the IoT scenario.
- **Chapter 2**: in order to measure its time performance, some researches were made to calculate timestamps delays in private and public LoRaWAN networks. Also to evaluate its feasibility, the well-known localization services (e.g., GPS and RTLS) integrate the LoRaWAN application in a smart city context. Finally, in other research line, LoRaWAN is used as a backup network for industry 4.0.
- **Chapter 3**: after a deep investigation and validation in LoRaWAN architecture in chapter 2, new researches were addressed and evaluated to enhance LoRaWAN technology. First, an innovative LoRaWAN mechanism is proposed and tested for extending LoRaWAN coverage, using an e-Node device, in an Industrial IoT scenario. Moreover, a novel LoRa-REP access method (inside e-Node, the same device used to increase LoRaWAN coverage) validates a redundant transmissions strategy to improve success probability in emergency situations.
- **Chapter 4**: finally, to integrate the promise LoRaWAN solution with health applications, a LoRaWAN modem is used to monitor the position and motion type of the elderly. The node is equipped with: motion sensor, accelerometer within an inertial measuring unit or IMU to fall detection, and localization engines based on Global Navigation Satellite Systems (GNSS) and Ultra Wide Band (UWD) for outdoor and indoor positions, respectively.

Each chapter has its own introduction, figures, tables, formulas, references and conclusions separately in order to make the thesis easier to read and understand.

1 IoT Based Architecture applied to Medical Cyber-Physical Systems (MCPS)

As the part of initial research in health area, this chapter aims to investigate and address an IoT Communication Architecture using MQTT and AMQP protocols for cloud data exchange in Medical Cyber-Physical System (MCPS) applied to a sample use case (Neurodegenerative Disease Diagnosis). This MCPS has paved a solid baseline in Health System area for further researches in the IoT scenario.

1.1 Objectives

In this chapter, a communication architecture, inspired by recent Internet of Things (IoT) advances, is proposed for connecting prototype instruments to the cloud, to allow direct and real-time interaction between developers and instrument operators. Without loss of generality, a real-world use case is addressed, dealing with the use of transcranial magnetic stimulation (TMS) for neurodegenerative disease diagnosis. The proposed infrastructure leverages on a message-oriented middleware, complemented by historical database for further data processing. Two of the most diffused protocols for cloud data exchange (MQTT and AMQP) have been investigated. The experimental setup has been focused on the real-time performance, which are the most challenging requirements. Time-related metrics confirm the feasibility of the proposed approach, resulting in an end-to-end delay on the order of few tens of milliseconds for local networks and up to few hundreds of milliseconds for geographical scale networks.

The aim of a Cyber Physical System (CPS) is to integrate the computational capability of the cyber world with the real-world physical processes by means of computer networks. A CPS consists of many different intelligent entities that can dynamically interact with the environment and can cooperate and self-organize, to make autonomous decisions [1.1]. In the recent past, this abstract concept has been profitably moved into the industrial automation world, applying large distributed computing systems in all the steps of the manufacturing process, from the initial planning to the aftermarket, in order to improve the efficiency and lower the costs. The availability of smart objects, constituting one of the fundamental pillars of the Internet of Things (IoT) revolution, has permitted the collection of a previously inconceivable amount of data from all the sensors and actuators deployed in the field [1.2].

The same way smart objects of the Internet of Things (i.e., devices hosting computational and communication capabilities) surround our everyday life, smart medical devices are currently used to continuously monitor and improve the patients' health status. From a different point of view, they can be considered a specialized version of embedded systems; they comprise sensors that sense the different parts of the patient's body supporting diagnostic decisions by means of easy-to-use and informative operator interfaces. Very different bio-signals can be acquired and analyzed, as the electrocardiogram (ECG) [1.3,1.4], the electroencephalogram (EEG) [1.5], the electromyogram (EMG) [1.6] just to mention a few. Data provided by these instruments are subsequently complemented by physicians' observation and constitutes a control loop with man-in-the-middle, having the medical workflow as the desired output.

As a consequence, despite several additional challenges in terms of security, dependability, privacy and safety exist when moving in the medical context, the idea of Medical CPS (MCPS) emerged as well [1.7–1.12], thanks to recent trends/needs in the medical environment.

Unfortunately, during early development stages, when only prototypes have been realized, the hardware, the internal firmware, and the procedure to use the medical instruments are not stable. Thus, MCPS medical communication busses cannot be used, due to the costs for making the instrument compliant with well-established standards. Complex certification and testing procedures must be repeated every time something inside the instrument is changed. Moreover, the MCPS communication systems have been created for satisfying the need for real-time communication on the local scale (e.g., in the intensive care units or operating rooms).

In the recent past, it was developed a prototype instrument for early neurodegenerative disease diagnosis [1.13, 1.14]. The need for a IoT-like real-time platform for connecting several prototypes and allow remote interaction arose during the clinical trials. The complexity of the test procedure is reflected by the many parameters that can be set by the operator (usually a Medical Doctor rather than a technician). Consequently, operator expertise can greatly affect the test result, limiting the adoption of the instrument itself. When prototype instruments are sent to other research teams for test, direct interaction on a geographic scale can facilitate proper instrument usage.

The main original contributions of this chapter are:

- the proposal of a novel IoT-based architecture addressing requirements of prototype medical instruments, with the following characteristics: low-cost; based on well-accepted, secure, open and interoperable message oriented solutions; and able to include other information coming from ancillary sensors;
- the identification and selection of formal performance metrics for the characterization of the real-time behavior of the proposed architecture;
- the implementation of a testbed with both local and cloud servers, in order to provide close-to-reality results;
- an experimental measurement campaign for the characterization of the real-time behavior varying the location of the servers and the size of the exchanged information.

For the sake of completeness, Message Queuing Telemetry Transport (MQTT) and Advanced Message Queuing Protocol (AMQP) protocols, exploiting the publish-subscribe paradigm, have been used and compared by means of the proposed metrics. Availability,

security overhead, leakage and tampering of data are out of the scope and have not been considered. In order to clarify the Chapter 1 workflow, the **Figure 1-1** shows its flowchart.

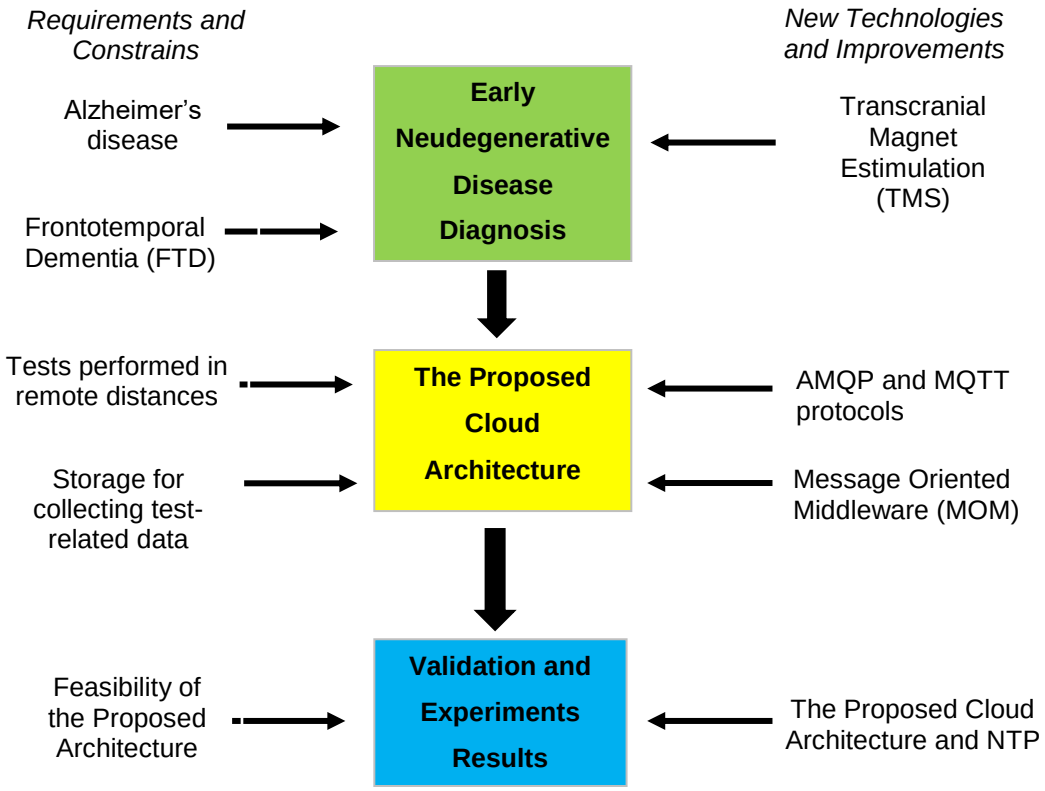


***Figure 1-1**: Graphical abstract of Chapter 1*

1.2 Early Neurodegenerative Disease Diagnosis

The aim of this section is to provide some background information about the prototype instrument of the use case. Alzheimer's disease (AD) and frontotemporal dementia (FTD) represent the main causes of neurodegenerative dementia in the population aged more than 60 years, with an estimated prevalence of 5% to 7%. Although it is quite easy to determine if a person is affected by dementia, early identification and classification of the specific disease is challenging (e.g., due to the fact that symptomatology is heterogeneous). Indeed, despite the AD is characterized by a well-established cholinergic impairment [1.15] and FTD is characterized by a GABA and glutamatergic impairment [1.15,1.16], very little has been done to develop a biomarker to identify a specific neurotransmitter.

1.2.1 Transcranial Magnetic Stimulation for Early Neurodegenerative Disease Diagnosis

However, it has been demonstrated that the impairment of different neurotransmitters, which AD and FTD selectively affect, can be effectively assessed by means of transcranial magnetic stimulation (TMS). In particular, cholinergic, GABAergic, and glutamatergic cortical circuits functionality can be indirectly evaluated by means of Short-latency Afferent Inhibition (SAI), Short-Interval Intracortical Inhibition (SICI), and Intracortical Facilitation (ICF) TMS paradigms, respectively [1.17 - 1.19]. This technique has several advantages, as this test is: a) non-invasive, reliable, easy to apply; b) not time-consuming, and c) suitable as a screening tool to differentiate AD from non-AD dementia with accuracy levels comparable to currently used techniques [1.13, 1.20].

TMS, originally proposed 1985, is a technique to study the motor cortex in a non-invasive and painless way; it takes advantage from an external magnetic field, penetrating the scalp and generating an induced current in the brain cortex, because of the Faraday law. Magnetic stimulus is created injecting a high current in a coil, resulting in a magnetic field up to 1–2 T, for a relatively short time (≈1 ms). In particular, the induced current allows to overcome muscle motor neuron threshold, resulting in patient's movement that is acquired by means of electromyography (EMG). Generally, the TMS excitation probe is located in proximity of the brain cortex region, where hand muscles are considerably represented, causing a movement of patient's fingers after a magnetic test stimulus (mTS). An adequate calibration procedure in terms of magnetic field amplitude, aiming at tuning the mTS amplitude, must be performed to obtain a patient-related reference baseline EMG response. Due to the length of the diagnosis procedure, the reference stimulus is typically repeated several times, to take into account possible baseline variations.

An mTS can be preceded by a magnetic subthreshold conditioning stimulus (mCS), sent in order to modify the patient's response to the mTS, depending on their time distance and the action of GABAergic and glutamatergic neurotransmitters. On the contrary, if an mTS is preceded by an electrical conditioning stimulus (eCS) up to 100 V, injected by electrodes located at the patient's wrist region, cholinergic impairment can be evaluated as well.

As a consequence, it is possible to define a stimulus train as the sequence of a conditioning stimulus followed by a test stimulus. When an mCS is followed by a mTS, a magnetic-magnetic (MM) train is defined; similarly, when an eCS is followed by a mTS, an electrical-magnetic (EM) train is defined. Finally, the reference stimulus (REF) is a particular stimulus

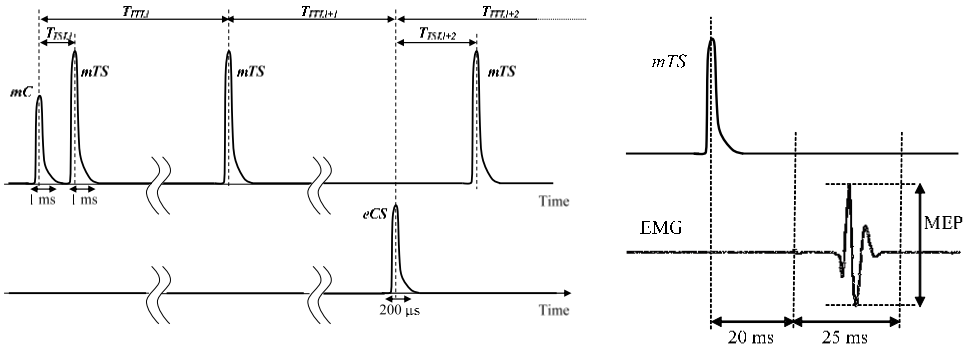
train, where only the test stimulus (mTS) is present. The time distance between a conditioning and test stimuli in a stimulus train is referred as the inter-stimulus interval, T_{ISI}. On the contrary, the time distance between the beginning of two consecutive stimulus trains is named inter-train interval, T_{ITI}.

1.2.2 Diagnosis Protocol with Transcranial Magnetic Stimulation

The diagnosis protocol is a random sequence of stimulus trains, each of which characterized by a proper type (REF, MM and EM), T_{ISI} and T_{ITI}. The definition of a diagnosis protocol implies the choice of the following protocol parameters:

- N_{MM}: the number of distinct values of T_{ISI} that should be used in each cycle of the diagnosis phase associated with MM trains;
- a set of N_{MM} T_{ISI};
- N_{EM}: the number of distinct values of T_{ISI} that should be used in each cycle of the diagnosis phase associated with EM trains;
- a set of N_{EM} T_{ISI};
- N_{REF}: the number of reference stimuli, i.e., REF trains, that should be issued in each cycle of the diagnosis phase;
- $T_{ITI,MIN}$, $T_{ITI,MAX}$: the range of T_{ITI} to use in the diagnosis phase, to ensure aperiodic stimulation for optimal patient response;
- N_C: the number of times that each cycle should be repeated in the diagnosis phase.

Resuming, each diagnosis protocol consists on $N_{TOT} = N_C (N_{MM} + N_{EM} + N_{REF})$ stimulus trains (see **Figure 1-2**a). Usually, $N_{MM} \leq 10$; $N_{EM} \leq 10$; $N_{REF} \leq 5$ and $N_{TOT} \leq 400$.



(**a**) (**b**)

Figure 1-2: (a) *Example of a set of three stimulus trains in a diagnosis protocol: train i is MM, train i + 1 is REF and train i + 2 is EM. The width of magnetic stimuli is usually 1 ms, whereas the width of electrical stimuli is 200 µs. TISI is usually between 1 ms and 100 ms, whereas TITI is on the order of few seconds.* ***(b)*** *Time diagram of mTS and successive EMG signal, with the quantity of interest (MEP).*

In both the diagnosis and calibration phases, the EMG should be acquired and processed to extract the quantity of interest, which is the peak-to-peak Motor Evoked Potential (MEP) measured in a time interval going from 20 ms to 45 ms after issuing the mTS, as shown in **Figure 1-2**b. The acquired MEP values are then clustered according to type and $T_{ISI,}$ and proper statistical analysis is performed to evaluate the patient reaction to different stimulations. Then allowing specialized medical personnel to identify and to classify possible presence of dementia disease.

Due to the novelty of the test procedure, there are no ready-to-use medical instruments but rather prototype systems must be developed. Some existing commercial devices can be adopted, as reported in [1.13], however, the setup flexibility of these solutions is somewhat limited and very often purposely-designed prototype devices are used [1.14]. In all cases, the expertise of the medical personnel conducting the test is very relevant in order to ensure affordable and reliable diagnosis. For instance, repeatability can be affected by improper magnetic probe and electrode positioning, improper calibration procedure and improper diagnosis protocol definition.

1.2.3 Advantages Offered by Cloud Services

As stated in the introduction, there are well-known advantages that can be obtained digitalizing all the possible information about the test procedure. However, the supporting infrastructure is one of the key point, as described in the next section.

Regarding prototype instruments, cloud services (like storage for collecting test-related data and computing for further processing) could speed-up the innovative diagnostic procedure validation and improve its diffusion among the medical community. Particularly, a single remote supervisor (e.g., one of the researcher who developed the TMS-based instrument) could receive measurement data from many instruments all around the world (see also example shown in **Figure 1-7**). More important, since a prototype device is neither standard nor widely diffused, the effectiveness of the test procedure could be greatly improved if it was possible for the supervisor to modify the diagnosis protocol in real-time, according to his/her knowledge of the instrument behavior. Consequently, end-to-end delays compatible with typical main-in-the-loop systems are key requirements for the communication infrastructure.

In Section 1.4, an overview of the proposed architecture is discussed and some time-related metrics are defined. In Section 1.5, experimental results are detailed.

1.3 The Proposed Cloud Architecture

In order connect the TMS-based prototype instrument to the cloud, some kind of "horizontal integration" must be provided, typically in the form of middleware. In particular, smart medical devices and operators should be able to enter and leave the infrastructure as desired. The typical approach is the adoption of a message oriented middleware (MOM) based on a publisher/subscriber protocol, in which nodes announce their new data and others can selectively subscribe to them. Some standardization efforts for providing a common platform have been carried out. In the past, the IEEE11073 has been created aiming at standardizing data formats for point-to-point connections of medical instruments and systems [1.21, 1.22]. Another example is the Integrated Clinical Environment (ICE) standard ASTM 2761-09, aiming at defining an interoperable environment for medical device integration, intended to improve patient safety [1.23]. However, despite an open source implementation exists, the OpenICE, based on the DDS—Data Distribution Service [1.24], its adoption is still limited, mainly due to the solution complexity. Indeed, proposals of simpler solutions exist, as the OpenICE-lite [1.25], leveraging the simpler Message Queuing Telemetry Transport protocol (MQTT). On the other hand, this approach suffers from the poor capability of the MQTT to customize data forwarding to subscribers and the need of an additional layer for secure transactions. These limitations are overcome using different protocols; since traceability of the test is important in medical applications and the number of nodes is relatively small, the Advanced Message Queuing Protocol (AMQP) has been identified as a viable approach; it offers higher flexibility in managing routes from the publisher to the subscriber and it is well suited for securely tracking transactions [1.22]. Put in other words, AMQP and MQTT are preferred to DDS due to their diffusion and easiness of implementation, that match well the needs for the prototype medical device addressed in this chapter. In the following, after a preliminary resume of protocol features, the proposed architecture is described and finally, some metrics for evaluating the performance are addressed.

1.3.1 The MQTT and AMQP Protocols

Both MQTT and AMQP are considered "message-centric" middleware, differently from the DDS, which is "data-centric". Very roughly speaking, data-centric middleware offers better scalability at the cost of increasing the complexity of the middleware; furthermore, data-centric systems are well suited for complex integration problems.

The MQTT protocol is an international standard (the ISO/IEC 20922), originally proposed for exchanging machine-to-machine telemetry data in low bandwidth environments. MQTT

relies on a message broker, which is a trusted intermediate in charge of ensuring message delivery from the publisher to the subscriber. Despite the acronym includes the term *Queuing*, the broker clients subscribe and publish on topics, that do not need to be previously created. Data are opaque to the broker; only the message topic is used for filtering and forwarding to the interested subscribers. The broker does not implement any store-and-forward mechanism, but offers three data delivery models corresponding to different message reliability and Quality of Service (QoS) levels. In particular, if QoS Level 0 is used, messages can be lost since there is no check of correct reception. If QoS Level 1 is chosen, a 2-way handshake mechanism is adopted, to ensure that each message is sent at least once. The QoS Level 2 implements a 4-way handshake, so that each message is sent exactly once. As a consequence, different QoS levels imply different delivery time. Regarding security concerns, MQTT relies by default on plain TCP transport protocol; however, many MQTT Brokers supports Transport Layer Security (TLS) for ciphering transactions and Simple Authentication and Security Layer (SASL) for authentication.

The AMQP-1.0 is currently an international standard (the ISO/IEC 19,464), that actively coexists with the AMQP-0.x implementations, due to the large number of available (open-source) solutions that still support the latter. In this chapter, without any generality loss, the AMQP-0.9.1 is considered and its main features are briefly discussed in the following. Interesting to note, the AMQP specifications include both the AMQ model, describing the components of an AMPQ-based middleware and their relationship, and the actual protocol that allow the clients to talk with the servers. In the AMQ model, the publisher sends the message to the “exchange” (a sort of mailbox) and the content of each exchange is delivered to one or more “queues” according to rules named “bindings”; the consumers subscribe to queues they are interested in. The virtual link between a publisher-subscriber pair is named “channel”. Message data are opaque to the broker (as for MQTT), except for some attributes the bindings rely on. This model makes the AMQP solution more flexible than MQTT, since the routing can be customized by the application simply changing the bindings. The QoS is related to the notion of message acknowledge, that can be automatic or manual. In the former case, the message is considered successfully delivered immediately after it is sent out (relying on the underlying transport layer for safe data delivery); in the latter, an explicit subscriber acknowledgement is required. In order to limit the traffic generated by acknowledges, a sliding-window mechanism can be implemented for manual mode, by means of a bounded channel pre-fetch threshold, that limits the number of ongoing message deliveries. Regarding

the connection security, AMQP is natively based on TLS for traffic encryption and SASL for authentication.

1.3.2 The Proposed Architecture

The general architecture of the proposed system for improving the effectiveness of prototype medical devices, as the TMS instrument previously described, is shown in Figure 1-3. The MOM, no matter the actual protocol, acts as a bidirectional pipe between an arbitrary number of TMS instruments on one side and an arbitrary number of users on the other one. Obviously, these end-points are not related to the actual topology of the system, but derive from the main data streams they are involved into. In particular, the *sensors measurements and systems status* data stream originates from the TMS instruments and propagates towards the end users, whereas the *commands and configurations* data stream flows in the opposite direction. The role of the MOM is to provide the proper "one-to-many" routing between the publisher (who generates the data stream) and the subscribers (which are interested in the data stream). As previously stated, publisher/subscriber paradigm offers the high flexibility and scalability required by the possibly very heterogeneous and dynamically-changing devices present in the considered scenario.

Additional services as caching (i.e., the capability to survive to network failures, as subscribers or network unavailability) and authentication and authorization (i.e., the capability to admit only trusted subscribers suitably identified) can be provided by the MOM itself or purposely-implemented as additional plugins or wrappers.

However, although the architecture shown in **Figure 1-3** is well-suited for machine-to-machine communications, it could be cumbersome for users (e.g., medical doctors) to access data adopting plain MOM clients. Human beings are used to accessing machines via browser-based applications; for this reason, a web server directly connected with the broker is present as well. The bidirectional communication on the user side actually occurs via a websocket, thus simplifying the management of asynchronous exchanges and limiting the overall overhead and the number of TCP/IP connections per user. Websockets overcome the request/response mechanism of HTTP and permit a persistent, asynchronous connection between the client and the server. In other words, websockets start with a regular HTTP connection and then mimic a raw TCP/IP exchange in the web context.

Figure 1-4 and **Figure 1-5** show the detailed architecture of the proposed framework when the AMQP and MQTT protocols are considered, respectively. For sake of generality, message storage is provided outside the broker by means of an additional database (DB), generally

located in the cloud. Similarly, an authentication/security manager is in charge of handling keys and certificates. As stated before, AMQP and MQTT leverage queues and topics for relaying messages to consumers; per each TMS instrument, a set of three queues/topics is provided: the DBQueue/Topic is for storing all the message in the storage DB; the WSQueue/Topic and the WSQueueR/TopicR are for websocket exchanges. In this chapter, MQTT is considered as a reference implementation, due to its simplicity and wide diffusion; AMQP is considered as the actual viable solution, due to its higher flexibility and scalability.

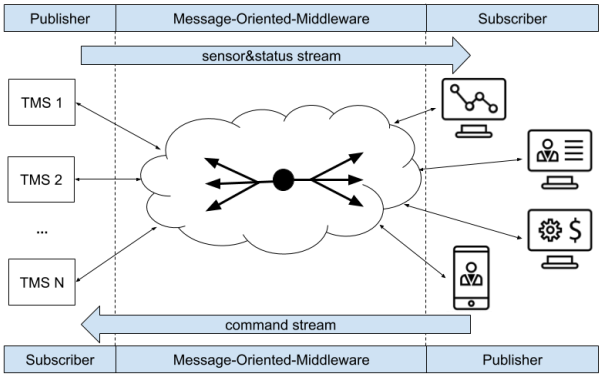


Figure 1-3: *The general problem of data exchange between TMSs and their authorized users.*

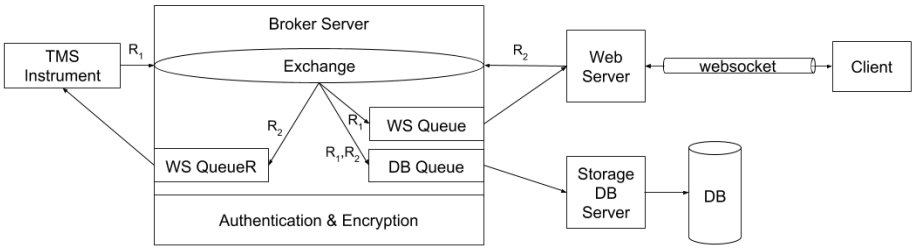


Figure 1-4: *Architecture of the AMQP-based solutions. The Authentication & Encryption tier is natively supported by the AMQP protocol.*

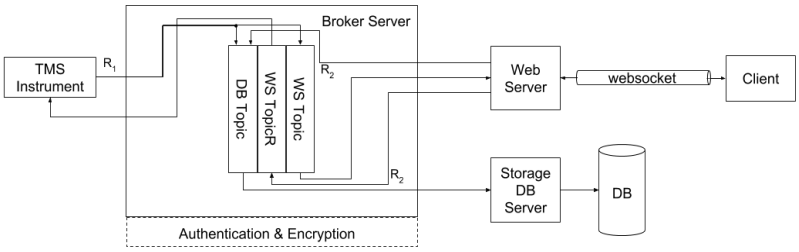


Figure 1-5: *The architecture of the MQTT-based solutions. The Authentication & Encryption tier has to be implemented at higher level since it is not directly managed by the MQTT Broker.*

In order to assess and verify the performance of a communication infrastructure, a suitable set of metrics must be defined. For instance, in the past, some efforts were carried out for measuring the Internet infrastructure at the network level [1.26]; metrics typically considered

include end-to-end or OWD (one-way delay) and delay variations [1.27, 1.28]. Referring to the Internet characterization, the RFC2679 defines the OWD as the time elapsing from the occurrence of the first bit of a packet at the source and the occurrence of the last bit of a packet at the destination; the RFC3393 introduces the Inter Packed Delay Variation (IPDV, i.e., the jitter), computed as the difference between the OWD of a selected pair of packets in a test stream. However, since the actual timing requirements and constrains depend on the application, the delay evaluation (and possible control) should be performed at the application level [1.29– 1.35].

This chapter is mainly focused on the real-time availability of data regarding the medical device; in this way, the information coming from the prototype instruments will be transparently shared among local and remote users. For this reason, a set of metrics related to the end-to-end delays introduced by the proposed architecture are defined. The delays can be calculated only if precise timestamps are taken at relevant events, like data/command transfers. Hence, the following timestamps are defined (see **Figure 1-6**):

- **T1**: this timestamp is assigned by the instruments when a new set of data is ready to be sent to the architecture.
- **T2**: this timestamp is assigned by the architecture when a new set of data arrives to the Web Server coming from the instrument.
- **T3**: this timestamp is assigned by the database when a new set of data coming from the instruments is permanently stored in its data table.
- **T4**: this timestamp is assigned by the Client when a new set of data arrives through the websocket.
- **T5**: this timestamp is assigned by the Client when a new command leaves the Client addressed to the architecture.
- **T6**: this timestamp is assigned by the architecture when a new set of data arrives to the Web Server coming from the Client.
- **T7**: this timestamp is assigned by the database when a new set of data coming from the Client is permanently stored in its data table.
- **T8**: this timestamp is assigned by the instrument when a command arrives through the architecture.

Using these timestamps, the proposed consistent set of metrics is:

- **DIC** = T4–T1: the overall end-to-end delay from the TMS instrument to the remote Client.
- **DIS** = T2–T1: the end-to-end delay from the TMS instrument to the Web Server.

- **DID** = T3–T1: the end-to-end delay from the TMS instruments to the Database.
- **DCI** = T8–T5: the reverse overall end-to-end delay from the remote Client to the TMS instrument.
- **DCS** = T6–T5: the end-to end delay from the Client to the Web Server.
- **DCD** = T7–T5: the end-to-end delay from the Client to the Database.

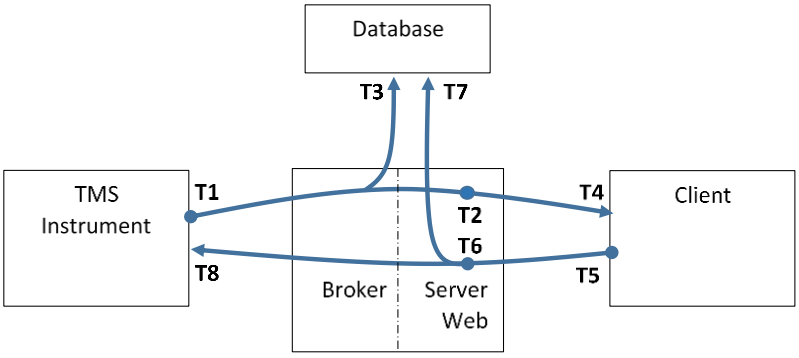


Figure 1-6: *The different timestamps collected for evaluating the proposed time-related metrics.*

The feasibility of the proposed architecture is investigated in this section. The different actors identified in the previous section are “black boxes” from the user’s point view; accordingly, the overall metrics of interest can be evaluated creating simulated scenarios [1.36, 1.37] or using specific experiments, as in [1.38, 1.39]. The latter approach has been pursued in this chapter. The used instruments and the experimental setup are introduced and characterized in specific subsections. Last, the experimental results are presented and discussed.

1.4 Validation and Experimental Results

The feasibility of the proposed architecture is investigated in this section. The different actors identified in the previous section are “black boxes” from the user’s point view; accordingly, the overall metrics of interest can be evaluated creating simulated scenarios [1.36, 1.37] or using specific experiments, as in [1.38, 1.39]. The latter approach has been pursued in this chapter. The used instruments and the experimental setup are introduced and characterized in specific subsections. Last, the experimental results are presented and discussed.

1.4.1 The Prototype Medical Instrument for Neurodegenerative Disease Diagnosis

As stated before, the aforementioned TMS-based technique for neurodegenerative disease diagnosis is currently applied by means of purposely-designed and ad-hoc implemented

excitation/acquisition systems. According to the operating principle, the minimal requirements include a magnetic and electric stimuli generator and an EMG acquisition system.

The magnetic excitation probe, suitable for cortical stimulation and monophasic waveforms, typically has a figure-of-eight shape and hosts a pair of coils, each of which with a diameter on the order of 100 mm. The operator handhelds the probe, so that the center is in contact with the patient's scalp and the coils are tangent with the scalp surface. The operator is in charge of ensuring the alignment with the motor cortex gyrus, i.e., the correct positioning of the coil involves movements in the antero-posterior and latero-lateral plains as well as rotation around the axis perpendicular to the scalp surface (typically 45° from the midline). As previously described, magnetic induction field is up to 2 T at the probe surface.

Bar electrodes are used for injecting electrical stimuli. They are typically applied at the patient's arm, few centimeters apart from each other. In particular, the bar electrodes should be placed at the wrist's internal side, over the median nerve, longitudinally oriented with the cathode positioned proximally.

Finally, MEP signals, obtained via surface EMG, are collected by means of single pair of Ag/AgCl electrodes, that should be placed on the first interosseous muscle, dorsal side, and metacarpal-phalangeal joint, lateral side, respectively. A third electrode should be used to provide a proper reference voltage to the user's wrist, dorsal side.

Magnetic and electrical actuating elements are connected to stimulators (drivers), whereas EMG sensing elements are connected to an amplifier. The magnetic stimulator should be able to issue REF stimulus trains (i.e., a single mTS) as well MM stimulus trains (i.e., an mCS followed by an mTS). Moreover, in case of EM stimulus trains (i.e., an eCS followed by an mTS), the magnetic stimulator takes care of generating the mTS, suitably synchronized with the eCS. The amplitude of the mTS and mCS stimuli should be independently tunable in step of 1% with respect to the maximum value. The electric stimulator should be able to issue the single electrical unipolar pulse eCS for the EM stimulus trains. The electrical stimulus should have a constant width of 200 μs (with a resolution of 10 μs) and magnitude should be tunable between 0 and 100 V in steps of 1 V. T_{ISI} interval, for both MM and EM stimulus trains, should be tunable between 1 ms and 250 ms in steps of 1 ms (with a resolution of 100 μs). The inter-train interval T_{ITI} should be tunable in steps of 100 ms, with a typical range from 4 to 10 s. The EMG amplifier should have an input range of at least 10 mVpp and a sampling frequency of at least 1 kHz (the minimum MEP bandwidth is typically 10 Hz–500 Hz); vertical resolution should be 10 μV (i.e., Effective Number of Bits—ENOB—10 bit). The EMG amplifier should be able to provide a proper ground voltage by means of the reference electrode. Commercial

drivers and amplifiers are available; in this chapter a BISTIM^2 system (from Magstim, Whitland, UK) for magnetic stimulation and a STMISOLA (from Biopac, Goleta, CA, USA) for electrical stimulation are used; a QUATTRO device (from Otbioelettronica, Torino, Italy) has been employed as electromyograph.

A controller unit is needed to manage all the blocks of the system. The controller is in charge of: correctly configuring and synchronizing the stimulators, to provide the proper sequence of stimuli (the diagnosis protocol); processing raw EMG data; and managing the web-based user interface for the local operator. In particular, the stand-alone operation of the instrument requires a minimum set of functionalities, including: the selection of operation mode (among diagnosis, magnetic calibration, electric calibration); issuing of start and stop commands for diagnosis and calibration phases; real time visualization of MEP value for the calibration phases; tuning of magnetic and electrical stimuli amplitude; visualization of the automatic response of data analysis after the diagnosis phase; and stimuli randomization.

Some of the authors developed a prototype controller, the core of which is based on a single-board computer running Linux (a Raspberry Pi board, chosen due to the compactness, computational capability, peripheral availability, and wide support from the user community). On the contrary, the synchronized management of the stimuli trigger signals (according to the diagnosis protocol previously described) has been assigned to a more deterministic Field Programmable Gate Array unit (FPGA, in particular a Cyclone III EP3C16F from Altera, San Jose, CA, USA).

In this chapter, the controller has been enhanced to further support the IoT-like framework detailed in Section 1.4. In particular, the system is able to provide raw data about the diagnosis protocol via AMQP (and MQTT) protocol. According to the previous statements, the useful information includes:

- protocol invariant data, as the amplitude of the stimuli, A_{mTS}, A_{mCS}, A_{eCS}; each of which is one byte wide;
- an array of N_{TOT} records, each of which specifying the stimulus type (1 byte), the T_{ISI} value (1 byte), the T_{ITI} value (1 byte), and the calculated MEP value (2 byte).

Since $N_{TOT} \leq 400$, the acquisition of a whole diagnosis protocol requires less than 2 KB. Additionally, the system is able to provide raw samples of the EMG signal, i.e., 25 samples 2 byte wide per each cycle; the overall EMG data require about 5 KB.

For sake of completeness, in **Figure 1-7** a possible dashboard illustrating data available after a diagnosis protocol has been completed is shown. As stated in the introduction, the proposed

solution allows a remote supervisor to interact in real-time with TMS-based instrument located all around the world. In particular, the current measurement results are presented in real-time in the "Show Result" page, whereas the supervisor can change the diagnosis protocol to be applied in the "Edit Protocol" page.

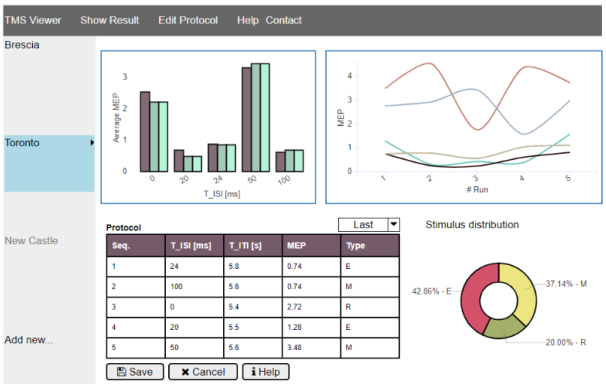


***Figure 1-7**: Client dashboard for remote supervision of diagnosis protocol carried out with the*

1.4.2 Experimental Setup

The experimental setup is depicted in **Figure 1-8**. There are five main actors involved in the experiments. The designed experiment requires a message starting from the TMS Instrument to travel toward the Client, being saved also in the database; in the Client, a loopback mechanism triggers the sending of a command message that travels back to the TMS Instrument, activating the saving action inside the database as well. AMQP and MQTT Brokers are implemented using open source solutions, RabbitMQ (version 3.7.8) and Mosquitto (version 1.5.3), respectively. They have been chosen due to their diffusion and availability for many different platforms. RabbitMQ is written in Erlang and has been developed within the Open Telecom Platform framework for failover and clustering. RabbitMQ natively supports AMQP-0.9.1 as the core protocol and offers a gateway for HTTP enabling a management plugin, mainly for diagnostic purposes, other than supporting messaging services toward a browser based on STOMP over websocket and JSON-RPC. Mosquitto is an Eclipse project; it implements MQTT version 3.1.1 and it is written in C. It has been mainly designed for satisfying lightweight messaging applications, especially when computational capabilities are scarce.

Each actor in the experiment has its own characteristics, as listed in the following:

- The TMS Instrument is based on a Raspberry Pi 2 single board computer, as described in Section 1.5.1. The raspberry Pi 2 hosts a quad-core 64 bit ARM processor (clock frequency 1.2 GHz), 1 GB of RAM, four USB ports, a LAN interface and HDMI video

connection; it runs the Linux-based Raspbian OS. The TMS Instrument role is to initiate the experiment, sending the first data message. It also terminate the experiment, receiving the Client command. A custom transport layer abstracting the communication libraries for both AMQP and MQTT has been implemented in Python; in particular, the transport layer uses the Paho library for MQTT and the Pika library for AMQP.

- The Client represents the remote operator that needs to interact with the prototype TMS Instrument. In this experiment, the simple Client role is to receive data from the instrument and loopback them into a command for the instrument. It has been implemented with an embedded device to introduce the minimum overhead. The device is a Siemens IOT2040 (based on an Intel Quark × 1020 (+Secure Boot) processor and hosting 1 GB of RAM, two Ethernet ports and two RS232/485 interfaces, and a battery-backed Real Time Calendar. It runs a customized Yocto Linux distribution. The entire software at client side is written in Javascript and runs in Node.Js, a well-accepted approach in IoT solutions that ensures portability on different platforms. The Client is located inside the domain of the University of Brescia.
- The Local Server is a machine specifically created for the experiments. It is located inside the domain of the University of Brescia. It is a VMware (ESXi 6.5) virtual machine (VM); the VM is hosted on a DELL PowerEdge R630 (Intel Xeon E5–2 core, 128 GB RAM, 4 × 1 Gbps Ethernet), one of the facility available in the University of Brescia eLUX laboratory [1.40]. The VM has a single CPU, 2 GB of RAM and has CentOS7 as guest operating System (OS). It hosts the RabbitMQ AMQP broker, with the HTTP management plugin enabled, and the Mosquitto MQTT broker. The local server hosts also the websocket server, which is a simple Python 3.6 program. It uses the Tornado web server websocket implementation to communicate with the Client and it runs the same custom transport layer used in the TMS Instrument (which, as previously stated, leverages the Paho library for MQTT and the Pika library for AMQP).
- The Cloud Server is an instance of the Amazon AWS EC2 t2.micro virtual machine with a single CPU and 1 GB of RAM. The Cloud Server is hosted by the US East (Ohio) Amazon Web Service (AWS) data center and it has been specifically created for the experiments. The software configuration of the Cloud Server is identical to the Local Server one; a minimal effort is required for migration from Local to cloud Server.

- The Database Server is located inside the domain of the University of Brescia. It is a VMware (ESXi 6.5) VM; it is hosted on a Syneto Ultra 205 hyper converged system (Intel Xeon E5–6-core, 64 GB RAM, 4 × 1 Gbps Ethernet), another facility available in the eLUX laboratory. The VM has 4 CPU, 16 GB RAM and has CentOS7 as guest OS. It hosts the different databases (including MariaDB, Influxdb, PosgreSQL) that collect the laboratory data. For the experiments of this chapter, the Influxdb, an open-source database purposely-designed for time series storage, has been chosen.

The experimental setup is easily configurable to change the message protocol (AMQP or MQTT) and the location of the Server. As shown in **Figure 1-8**, the AMQP scenario is based on the AMQP blocks inside the actors, with the dashed lines showing the message exchange paths. On the other hand, the solid lines connect the MQTT blocks for the implementation of the MQTT scenario. The local and cloud scenario are denoted by different line color: orange for Local Servers, blue for Cloud Server. In all the experiments, default configuration of Brokers is considered, if not explicitly stated otherwise; due to the previously highlighted constraints, message size is set to 2 KB, representative of typical information transferred per diagnosis protocol, as stated in Section 1.5.2.

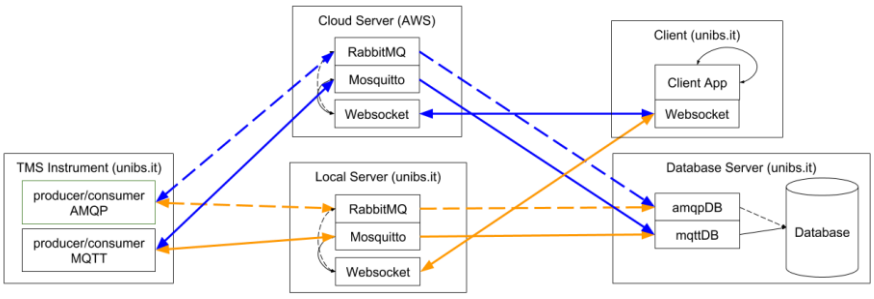


Figure 1-8: Block diagram of the implemented experimental setup

1.4.3 Estimation of Measurement Uncertainty

To calculate the metrics defined in the previous section, a synchronization mechanism between all the devices/machines involved in the distributed system is necessary. In this chapter, the NTP synchronization is used. It is periodically executed in every device and the local system clock is corrected using the offset and delay measurements from a global NTP server, which is, in turn, synchronized to the UTC source. The uncertainty of the time synchronization using NTP varies with the quality of the Internet connection in terms of latency, and it is typically on milliseconds scale [1.41] The NTP daemon transparently calculates and applies corrections, logging error statistics when requested. In this chapter, the synchronization uncertainty has been evaluated using the residual offset of the system clock at

step N, which is the residual error after the clock corrections calculated at step N-1 have modified the local clock behavior. The results are shown in Table 1-1 and **Figure 1-9** for a one-day observation time, compatible with the observation time used to evaluate the experiments.

There is a systematic error introduced by the NTP software which is hard to be compensated by calibration procedures. The experimental synchronization standard uncertainty u_{sn} takes into account all the error components: the average error μ_{sn} and the standard deviation σ_{sn}. For this reason, the experimental synchronization standard uncertainty has been evaluated as $u_{sn} = \sqrt{\mu_{sn}^2 + \sigma_{sn}^2}$ [1.42].

From Table 1, the synchronization uncertainty (with respect to the UTC) of all the devices involved in the experimental setup is less than 0.3 ms. In the proposed experimental setup, the event timestamping is carried out in software. This situation introduced an uncertainty contribution which is evaluated with a specific experiment, the full description of which is given in [1.34] and briefly resumed here. The timestamp uncertainty is estimated executing a software loop that collects a timestamp just before (T9) and just after (T10) a routine that creates a fixed and known delay of value K. In theory, the timestamps are taken from the same system clock used to calculate the delay, so the quantity $\Delta = T10 - T9 - K$ should be equal to zero. In practice, the experimental results show a variability and the uncertainty u_Δ of the quantity Δ can be modelled as $u_\Delta = \sqrt{\mu_\Delta^2 + \sigma_\Delta^2}$, because both standard deviation σ_Δ and the average value μ_Δ (i.e., the systematic error) are considered. Supposing that the uncertainty u_Δ is completely due to the timestamp uncertainty u_{tn} that disturbs both T8 and T9, u_{tn} can be estimated with the Eq. 1.4.1:

$$\boldsymbol{u_\Delta^2 = u_{tn}^2 + u_{tn}^2 = 2u_{tn}^2 \;\rightarrow\; u_{tn} = \sqrt{\frac{u_\Delta^2}{2}},} \qquad (1.4.1)$$

The results (over 1000 samples) are shown in Table 2 for all the devices involved in the experiments; the maximum timestamp standard uncertainty u_{tn} is on the order of 3 ms for the Client machine, whereas all the other devices show uncertainty under 1 ms.

It has to be noted that: all the devices are able to synchronize with a very low average offset (Table 1), and the timestamp uncertainty is dominating over the synchronization uncertainty.

The metrics described in Section 1.4.3 involve the timestamps taken in different devices. The uncertainty u_M of each metric is different, depending on the uncertainty of the devices involved in its calculation. Equation 1.4.2 can be used to estimate the uncertainty of the

measurements. The resulting uncertainty values are reported in Table 1-3 for all the metrics to be calculated, when the synchronization uncertainty and the timestamp uncertainty of Table 1-1 and Table 1-2 are considered. In details, being A and B two devices used for the metric calculation, the relation expressing the combination of the uncertainties is the following:

$$\mathbf{u_M} = \sqrt{\mathbf{u_{sA}^2} + \mathbf{u_{tA}^2} + \mathbf{u_{sB}^2} + \mathbf{u_{tB}^2}} \qquad (1.4.3)$$

As expected, due to the squaring operation, the main contribution dominates and, analyzing Table 1-3, it is clear that the metric uncertainty is greater when a timestamp taken by the Client is involved in the metric calculation.

Table 1-1: *Synchronization uncertainty of the devices involved in the* EXPERIMENTS *(ms).*

Device	u_{sn}	μ_{sn}	σ_{sn}
TMS Instrument	0.009	0.000	0.009
Local Server	0.172	0.007	0.172
Cloud Server	0.278	0.013	0.278
DB Server	0.126	0.011	0.125
Client	0.018	0.001	0.018

Table 1-2: *Timestamp uncertainty of the devices involved in the experiments (ms).*

Device	u_{tn}	μ_{Δ}	σ_{Δ}
TMS Instrument	1.0	1.4	0.2
Local Server	0.7	1.0	0.1
Cloud Server	0.8	1.1	0.3
DB Server	0.8	1.0	0.3
Client	3.1	4.3	0.9

Table 1-3: *Synchronization uncertainty related to the metrics used in the experiments (ms).*

	Metric	**Device A**	**Device B**		u_M
Local	TMS to Client	TMS Instrument	Client		3.3
	TMS to Server	TMS Instrument	Local Server		1.2
	TMS to Database	TMS Instrument	DB Server		1.3
	Client to TMS	Client	TMS Instrument		3.3
	Client to Server	Client	Local Server		3.2
	Client to Database	Client	DB Server		3.2
Cloud	TMS to Client	TMS Instrument	Client		3.3
	TMS to Server	TMS Instrument	AWS Server		1.3
	TMS to Database	TMS Instrument	DB Server		1.3
	Client to TMS	Client	TMS Instrument		3.3
	Client to Server	Client	Cloud Server		3.2
	Client to Database	Client	DB Server		3.2

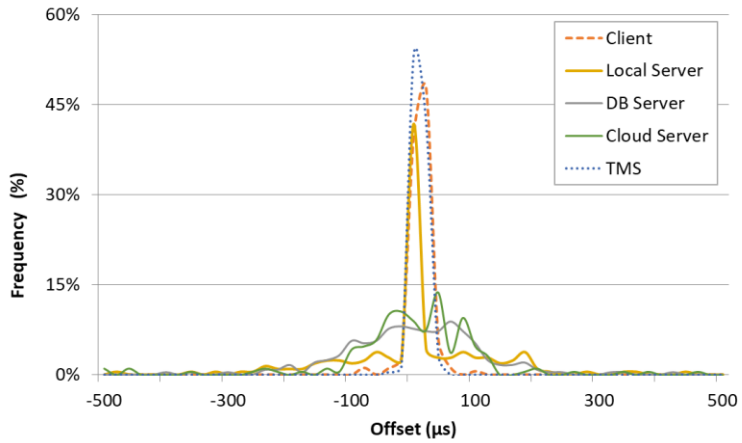


Figure 1-9: *Probability Density Function (PDF) estimate of the residual offset after NTP daemon corrected the local system clock of the devices involved in the experiments.*

1.4.4 Experimental Results

Each single experiment is aimed at collecting all the relevant timestamps along the path described in Section 1.5.2. More than 200 samples are collected for each experiment run. The experiment campaign is composed of eight scenarios, obtained varying the following parameters:

- interval between data transfer (1 s or 10 s);
- location of Broker and Web Server (Local Server or Cloud Server);
- the used message protocol (AMQP or MQTT).

The transferred data is 2 KB per message and the DB server and the Client are in the same local network. The metrics defined in Section 1.4.3 have been calculated for each scenario and are shown in Table 1-4.

Table 1-4: *Performance metrics of the proposed architecture as implemented for the experiments (ms).*

	Metric	**MQTT @ 1 s**		**AMQP @ 1 s**		**MQTT @ 10 s**		**AMQP @ 10 s**	
		Mean	**95 perc.**	**Mean**	**95 perc.**	**Mean**	**95 perc.**	**Mean**	**95 perc.**
Local	D_{IC}	4.8	5.4	7.0	7.6	6.3	6.3	6.2	6.5
	D_{IS}	0.5	0.9	2.0	2.3	1.7	1.8	2.0	2.3
	D_{ID}	2.3	3.5	3.8	4.3	3.4	4.1	3.8	4.4
	D_{CI}	25.7	30.9	27.8	34.6	26.2	33.5	25.0	29.8
	D_{CS}	22.1	25.4	24.1	30.6	22.7	26.4	21.2	25.2
	D_{CD}	25.4	30.7	28.6	35.5	28.2	37.3	25.9	30.8
Cloud	D_{IC}	144.6	158.9	132.5	134.6	145.0	164.3	141.8	143.0
	D_{IS}	71.4	71.9	59.8	60.3	71.9	72.9	72.0	72.7
	D_{ID}	141.9	147.7	128.1	129.0	141.1	148.9	140.3	141.9
	D_{CI}	164.3	189.4	168.5	176.7	165.9	186.3	164.0	183.3
	D_{CS}	93.7	111.8	97.5	103.8	94.1	110.6	94.5	115.3
	D_{CD}	164.3	189.5	169.3	178.2	165.8	186.8	166.5	185.5

First of all, it can be affirmed that the AMQP and the MQTT implementations practically introduce similar delays, even if the number of features offered by the two brokers is different. Additionally, from the previous results, it is clear that, when the cloud scenario is considered (messages passing twice across geographical networks, i.e., from the TMS instrument to the cloud server and then from the cloud server to the database server or the client), an additional delay of about 130 ms is added. Nevertheless, the 95 percentile of the end-to-end delay is always less than 200 ms, so that the proposed communication framework can satisfy real-time requirements of supervision/control applications with human in-the-loop. Finally, there is an asymmetry between the data flow (from the TMS instrument to the operator) and the commands flow (the opposite direction); it has to be attributed to the different behavior of the experimental websocket server managing input or output streams, since websocket sessions are mostly symmetrical.

For sake of completeness, the boxplots of the DIC and DCI performance metrics for different intervals between data transfer are shown in **Figure 1-10** and **Figure 1-11**, for local and cloud server implementations, respectively. The boxplots highlight the distribution of the samples and, in this specific case, it is evident the larger support of the distributions related to the cloud server (as expected). Last, it can be argued that standard deviation of the metrics is (often) greater than the measurement uncertainty, highlighting the additional variability introduced by the whole system under test.

Another experiment has been designed to understand the relation between the size of the exchanged information and the end to end delay. In this experiment, the DIC and DCI metrics have been considered in the case of cloud server (which is the most demanding). The results for size varying from 1 KB to 10 KB are graphically presented in **Figure 1-12**. As expected, the size of data has a direct impact on the DIC and DCI. The delay grow is higher for MQTT when the size increases, whereas the variability is almost the same between AMQP and MQTT.

In conclusion, considering all the experimental evidences, the AMQP solutions offers many more features (e.g., security and authentications) with respect to the MQTT basic implementation, without sacrificing the performance. For this reasons, AMQP seems to be the best solution for the implementation of the proposed architecture.

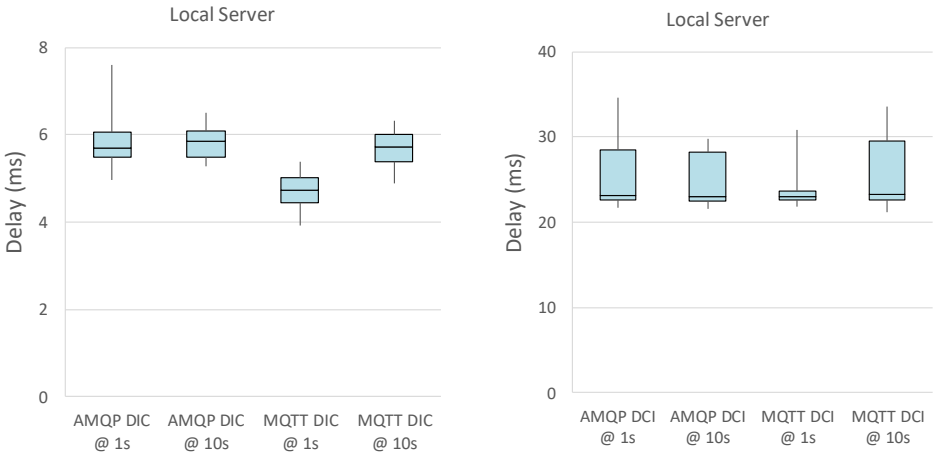


***Figure 1-10**: Boxplot of the DIC and DCI performance metrics with local server. The plot shows a blue box from the 25 percentile to the 75 percentile (with the median line). The whiskers go from the 5 percentile up the 95 percentile.*

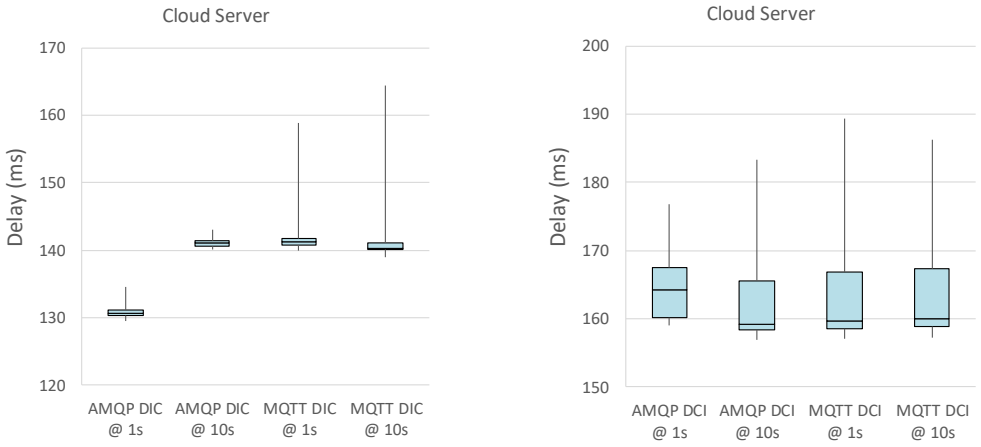


***Figure 1-11**: Boxplot of the DIC and DCI performance metrics with cloud server. The plot shows a blue box from the 25 percentile to the 75 percentile (with the median line). The whiskers go from the 5 percentile up the 95 percentile.*

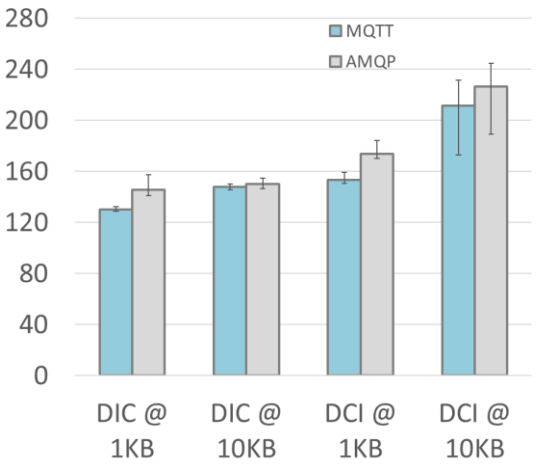


***Figure 1-12**: Bar graph of the DIC and DCI performance metrics with cloud server. The whiskers go from the 5 percentile up the 95 percentile.*

1.5 Final Considerations of IoT Based Architecture applied to MCPS

Cyber-physical systems are characterized by the close and tight coupling between the real-world physical components (with their own dynamics) and the digital counterpart aimed at monitoring and controlling. The interaction between the two domains requires completely new communication paradigms, which are changing the way humans interact with and control the surrounding environment. CPSs must satisfy stringent requirements in terms of dependability, security, safety, and efficiency in real time, while satisfying privacy constraints. Healthcare and medical devices and systems have been transformed into MCPSs, taking advantage of improved data sharing capabilities. In this chapter, inspired by recent IoT advances, a simple real-time solution to address peculiarities of prototype medical diagnostic tests, which are often based on proof-of-concept instruments requiring a high expertise level by the operators, is proposed. The focus is an innovative technique for early diagnosis of dementia based on Transcranial Magnetic Stimulation, but the proposed approach may be applied for different diagnostic solutions at the prototype stage. It was demonstrated that the use of a message oriented middleware allows to easy interconnect in real-time several prototype instruments and medical personnel on both local and geographical scale. In particular, when widely diffused and well-accepted protocols as AMQP and MQTT are adopted, it has been experimentally verified that the end-to-end delay of the communication framework is compatible with real-time requirements of medical control applications with a human-in-the-loop. Moreover, the preference goes to the AMQP solutions: the considered scenario, AMQP has the same performance of MQTT and it natively offers more features (like authentication). A purposely-designed experimental setup has been implemented to evaluate time-related metrics on both a local and geographical scale. Thanks to the proper design of the infrastructure, moving from local to geographical scale is almost straightforward. The synchronization capability has been verified, resulting in a worst-case synchronization uncertainty on the order of 3 ms. The overall end-to-end delay from the TMS instrument to the remote client has an average value of about 140 ms, which rises up to about 160 ms for the opposite direction, confirming the feasibility of the proposed approach. As a concluding remark, the experiments highlight that, when the geographical scale is considered, the cloud service provider may greatly impact on performance, no matter the peculiar implementation.

In the future, the proposed solution could be profitably used for exchanging not only raw test samples (to be further processed in the cloud), but ancillary meta-data as well (e.g., images or small videos of the testing procedure, real-time location of the personnel and patient involved in the test, serial/identification numbers of the adopted instrumentation, etc.). For

these reasons, specific plugins implementing typical MCPS interface (e.g., OpenICE) are under investigation.

The IoT based architecture applied to MCPS, using MQTT and AMQP communication protocols, has opened a solid baseline in Health Application systems for the next researches in the IoT scenario, as demonstrated in the following chapters.

2 LoRaWAN infrastructure

This chapter aims to make a deep investigation and characterization of LoRaWAN, the most promising wireless LPWAN technology. First, in order to measure time performance, multiple research campaigns was made to calculate transmission delays in private and public LoRaWAN networks. Moreover, well-known localization services (e.g., GPS and RTLS) were integrated in LoRaWAN applications for smart city in order to evaluate the feasibility. Finally, LoRaWAN is characterized as a backup network for Industry 4.0.

2.1 Objectives

The proposed MCPS investigated in the Chapter 1 demonstrates the feasibility and usability of the IoT paradigm integrated with health applications, but also exposed the constraints related to wired connectivity, limiting the system flexibility and also the patient mobility. In order to solve these limitations, wireless technologies are the best options to be applied to the health context. Obviously, not all wireless solutions satisfy the demands for eHealth applications over large areas in an IoT scenario. This chapter investigates and evaluate the newest and most promising LPWAN technology, the LoRaWAN network. The main studies conducted are related to:

- **Architectures Delays of LoRaWAN applications**: a test methodology valuates the experimental assessment of architectural delays of public and private LoRaWAN networks, using different implementations and several hardware platforms.
- **LoRaWAN applications in a Smart City context:** localization engines for indoor and outdoor positions are integrated with a LoRaWAN modem to evaluate its feasibility in real use case scenario.
- **LoRaWAN infrastructure for Industry 4.0**: a LoRaWAN modem is used as a backup network in case of lacking 3G/4G connectivity on tracking and diagnosis of smart vehicles on the road for Industry 4.0. This research aims to demonstrate the application constraints and design directions to find the best integration between LoRaWAN infrastructure and existing frameworks and concepts.

In order to clarify the Chapter 2 workflow, the **Figure 2-1** shows its Flowchart.

2.2 Architecture Delays of LoRaWAN applications

In order to obtain the best architecture delays outcomes in LoRaWAN applications, tests campaigns were performed using timestamps measurements in two types of LoRaWAN infrastructures:

- **Private LoRaWAN networks**: a test methodology for the experimental assessment of architectural delays of LoRaWAN networks, by means of Semtech reference and LoRa Server Open Source implementations, using several hardware platforms in a private network in University of Brescia/Italy. The new approach is composed of: a mapping phase, needed for highlighting the LoRaWAN blocks inside the implementation under test; followed by a measurement phase, in which the relevant timestamps are taken along the information path from LoRaWAN node to LoRaWAN customer application.

- **Public LoRAWAN network**: it is implemented a real use case based on a public LoRaWAN infrastructure from Brescia Smart Live (BSL) project, by means of Patavina NetSuite solution from A2A Smart City. The tests propose metrics to formally evaluate the delays at the application level (i.e. considering the whole infrastructure as a black box).

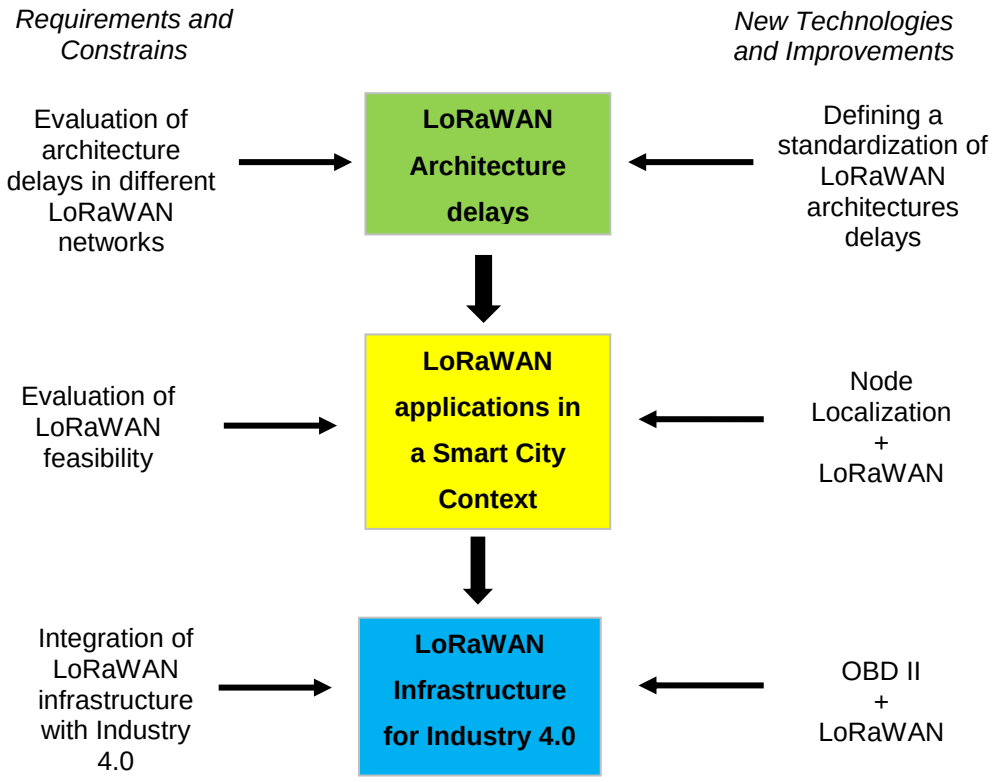


***Figure 2-1**: Graphical abstract of Chapter 2*

2.2.1 Private LoRaWAN Networks

Today, the availability at low cost of system-on-chip with enhanced performance and embedded connectivity boosts the era of connected smart objects to build the Internet of Things (IoT). The smart objects will soon change the way we live and work since their number will rise to billions in few years. Currently, the communication infrastructure is the Internet. However, many smart objects are not directly connected to the Internet but gateways are used to balance the costs. Last, if flexibility and mobility are required, wireless technologies are applied preferring low-cost and low-power consumption.

Among the various wireless technologies used to bridge the gap between sensors and gateways to the Internet, there are solutions inspired in the cellular communication approach. In these cases, the wireless end-devices are linked to a base station, which defines a one-hop cell; the base stations are connected together with a backbone network (wired or wireless).

However, despite current cellular network infrastructure technology (4G) could potentially address many IoT scenarios, the complex architecture and the demanding protocol stack result in high computational capability of the end-nodes, which in turn implies high-cost and high power consumption, something not compatible with most of actual applications. Only the next generation technology (5G) promises a mass deployment in the future. Therefore, other technologies are used today in this field: the Low Power Wide Area Networks (LPWANs) [2.1]-[2.4]. Their main advantages are the wide area coverage and the reduced number of base stations, thanks to: the innovative radios that trade off the sensitivity with the throughput; the simplified communication protocol stack that, in turn, reduces the infrastructure complexity. Consequently, both the power consumption and the cost are diminished [2.5], [2.6]. Currently, the consumer market, the industry, and the academic world are proposing/using several LPWANs; the 3rd generation partnership project (3GPP) itself standardized the so called "Narrowband-IoT" or NB-IoT in LTE release 13, focusing on machine-to-machine, delay-tolerant communications. Nevertheless, the use of licensed bands affects costs as well and makes device management cumbersome (e.g. due to SIM cards). At this time, most of the adopter consensus is achieved by LoRaWAN, that has been turned in a de-facto standard. It is an open standard defined by the LoRa Alliance, that now is implemented in many commercially available devices from multiple manufacturers. The LoRaWAN wide success is demonstrated by both the constantly increasing applications in various scenarios, and the huge available scientific literature.

Year after year, many implementations of the LoRaWAN specifications have appeared in the market, each one showing support for different hardware, for different backend software architectures, and for different operating systems. When many implementations are offered to network managers and end users, the need of test procedures for comparing them arises.

The scientific focus on LoRaWAN is on various aspects. There are papers providing discussions about theoretical limits imposed by the protocol, for instance medium access policies or duty cycle restrictions imposed by regulatory bodies [2.7]-[2.10]. There is an analytical model of the LoRaWAN uplink (UL) performance, in terms of collision rate and latency, under the influence of regulatory and aggregated duty cycling [2.11], with the hypothesis of exponential inter-arrival times. Finally, some works tried to experimentally evaluate communication performance [2.12]-[2.14]. The impact of quasi-orthogonal chirp sequence has been investigated as well [2.15]. The authors of this research are active in the

LoRaWAN area with works evaluating performance of the LoRaWAN wireless link [2.16]-[2.22].

In general, the most requested metrics for Quality of Service (QoS) evaluation in any business-critical applications are network latency and, especially, application response time. Nevertheless, end-to-end delay measurements and estimation technique for generic LoRaWAN implementations have not been presented in literature. In particular, discussion of appropriate methodologies, that can significantly improve quality, accuracy, and representativeness of measurements, is completely missing. What is an appropriate definition for LoRaWAN QoS metrics? How should each metric be measured? These questions currently have no answer.

This section tries to solve this gap, expanding the ideas presented in [2.23]. The central observation is that LoRaWAN specifications include a network reference model that all the compliant solutions must follow. Accordingly, some probing points for timestamping can be identified for any LoRaWAN application, independently of peculiarities of actual implementation. In particular, original contributions in this section are:

- Identification of logical entities comprised in the LoRaWAN network reference model, highlighting the communication "boundaries" (i.e. interfaces) among them;
- Identification of the different traffic flows existing between endpoints, focusing on the LoRaWAN backend services other than the wireless connectivity;
- Proposal of "universal" probing points for timestamping, and definition of metrics able to identify LoRaWAN network delays and to evaluate LoRaWAN implementation performance;
- Demonstration of the feasibility and the effectiveness of the proposed approach in real-world use cases leveraging on reference LoRaWAN implementations.

It has to be noticed that the proposed methodology is mainly aimed at evaluating the LoRaWAN QoS from the offered service point of view. Since backbone/backhaul network delays and backend server delays are "tunable" by choosing different implementation details (transport medium, bandwidth, etc…), the proposed methodology can help a service provider to easily compare and tradeoff features of different implementations.

2.2.1.1 IoT wireless networks: the LPWAN approach

In this section, a LPWAN general overview is given, and the LoRa solution is proposed as the most significant example of LPWAN radio technology currently available. In details, after a general description of the two lowest protocol layers (the physical layer PHY and the data

link layer DL) characteristics, the focus will return to the peculiarity of the LoRa radio developed by Semtech.

2.2.1.1.1 LPWAN technologies

The availability of wide coverage, the scalability, the low-cost, the low-power consumption of some wireless communication solutions are the key factor to make real and affordable many IoT applications. LPWAN has been proved to be the answers to these requests, if a relatively low update rate can be accepted. Generally speaking, technologies belonging to this class of wireless networks are characterized by:

- Operations in the different parts of the spectrum alternative to the over-crowded 2.4GHz region;
- Limited data rate, with small channel bandwidth, efficient modulation schema (for instance based on spread spectrum techniques), and a very good receiver sensitivity (allowing a link budget on the order of 150dB), and;
- Lightweight protocol stack and low complexity communication infrastructure, in order to minimize the power consumption and the cost.

The network is organized as a single-hop, star topology and medium access is done with random mechanisms like ALOHA [2.24]; the fulfillment of Regional regulations is usually accomplished using "listen-before-talk" and duty-cycle based strategies. With the aim of further reducing the costs of deployment and maintenance, the network complexity is mainly in the backend system while the end-devices are kept simple. The backend servers manage the network by assigning resources with the goal of satisfying individual application requirements. Several very different technologies have been grouped under the LPWAN umbrella 2.1. A very rough classification distinguishes between solutions dominated by technologies operating in license-free bands (typically using proprietary physical layer like the Ultra Narrow Band by SIGFOX, the LoRa by Semtech, and the Weightless led by Neul) versus technologies operating in licensed portions of the spectrum (most notably, the Narrow-Band IoT proposed in the 3GPP Long Term Evolution standard).

Additionally, it has to be remarked that in the last years the development and first experimental deployments of the 5th generation of mobile networks (5G) are happening. The main goal of 5G is to deliver huge capacity and scalability with respect to 4G mobile solutions. Consequently, the LPWANs and 5G solutions may operate together, with 5G as the backbone

connectivity of LPWAN. In some papers, such possible integration has been discussed [2.25],[2.26].

However, today, among all the recently appeared LPWAN technologies, the most interesting is still LoRaWAN that uses LoRa at lower levels. As a matter of fact, many manufacturers offer LoRa based commercial devices and modules, resulting in a large acceptance by both market and academic world. For the same reason, very different application scenarios have been considered as well, ranging from precision agriculture [2.28] to electric vehicles management [2.19],[2.29], and, finally, industry [2.18],[2.27].

2.2.1.1.2 The LoRa approach to spread spectrum

The LoRa radio has been patented by Semtech, and it is used in many communication stacks (including the LoRaWAN solutions) [2.30]. LoRa is based on Chirp Spread Spectrum (CSS) modulation that codes, with a single chirp frequency trajectory (i.e. a symbol), SF (spreading factor) bits. The choice of CSS has been done because the chirp symbols have good correlation properties and, changing the SF factor, virtual channels or adaptive data rate strategy can be achieved. In details, the chirp bandwidth BW is fixed (BW = 125, 250 or 500 kHz), while the chirp duration can be calculated as $T_C = \frac{2^{SF}}{BW}$. Accordingly, a higher SF means a lower data rate but a better noise immunity (thanks to additional processing gain). There is a Forward Error Correction mechanism that further increases noise and interference immunity, but in turn, it affects (reduces) the actual throughput; however, several coding rate can be specified in the range from CR=4/5 to CR=4/8.

It should be noted that the sub-GHz band must respect Regional regulations, including additional limitations on duty-cycle (that is the ratio between the duration of a transmission and time between transmissions). In Europe, for instance, the duty-cycle in the 868MHz band is (just) d=1% on all the three mandatory channels.

As other LPWAN solutions, LoRa may be connected to medium access control layer of several stacks, even if simple approach is usually preferred (e.g. ALOHA or clear channel assessment for limiting collisions in dense environment).

Concluding, it has to be stressed that, being the LoRa technology one of the most affordable technologies for implementing IoT applications requiring long range wireless connectivity, many literature works exist assessing its performance and its limitations, mainly imputable to the relatively poor throughput and bandwidth availability [2.7]-[2.15], [2.22]. A tradeoff between the number of sensors deployed in the field, the chosen spreading factor and the transmission rate exists. Indeed, for all those applications requiring one packet per day

transmission, millions of sensors are manageable; on the contrary, a refresh rate on the order of one minute lowers the number of possible nodes to thousands [2.17].

2.2.1.2 The proposed Methodology for Evaluating LoRaWAN solutions

A complete communication solution has been designed around the LoRa physical layer, named LoRaWAN. It has been designed in order to fulfill typical requirements of IoT scenarios leveraging autonomous (e.g. battery-powered) end-devices that may be either mobile or mounted at a fixed location and require relatively long refresh rate.

LoRaWAN specifications provide additional restrictions on the LoRa radio depending on the so called Regional Parameters. In Europe the allowed bandwidth is BW $\in$ [125,250] kHz and SF $\in$ [7..12]. Thus, each physical channel can host up to six quasi-orthogonal virtual channels (one per SF value), with data rates ranging from about 300 bps to 11 kbps. The effect of local clock drift in the nodes is limited keeping the message length in the range from 51 byte (at SF=12) to 242 byte (at SF=7).

In this section the architecture of a general LoRaWAN solution is described and, starting from the resulting model, the proposed approach for assessing architecture performance is discussed.

2.2.1.2.1 The LoRaWAN architecture

LoRaWAN specifications, which are publicly available, are managed by the so called LoRa alliance, that includes both device manufacturers, end users and research institutions. In particular, LoRaWAN completes the communication protocol stack defining the data link layer, that implements an ALOHA like medium access strategy. Regarding the network level, the addressed network topology is a hybrid wireless and wired star-of-stars, with the aim of minimizing as much as possible complexity (and thus implementation and maintenance costs) on the wireless side. Such an approach mimics the well-known topology of mobile communication networks, where handsets are wirelessly connected to a base station (thus forming a wireless star) and several base stations are interconnected to a base station controller using wired connections [2.31]. LoRaWAN specifies that a network comprises gateways (also known as concentrators or base stations) that relay end-device messages to the backend; interesting to note, each gateway contains a software (called "*packet forwarder"*) that is able to forward messages using an implementation-specific protocol. It has to be highlighted that the gateway operates only at the physical layer and the frame payload is opaque.

In other words, two tiers exist in the network; one is based on wireless links to provide connectivity to end devices; the other is the backend, where network management is carried out in a centralized way and services to end-users are provided. The actual data exchange consists of: uplink, when the communication occurs from the end-device towards the user; and downlink, when the opposite occurs. According to LoRaWAN target applications, uplink is of main importance and concern, whereas downlink should be limited to increase efficient bandwidth utilization [2.32], [2.33].

More in detail, the network reference model described in the LoRaWAN specifications [2.34] consists of two or three different kind of *servers* (depending on the version compatibility): the Network Server (NS); the Application Server (AS) and the Join Server (JS), as shown in **Figure 2-2**. Note that the implementation details are out of the scope of the specifications, while only the operations to be carried out are described.

The NS is the logical entity that resembles the center of the star topology; the NS is in charge of checking proper frame format and authentication, providing acknowledgements if required. Additionally, the NS actually manages the LoRaWAN data link protocol features, e.g. as data rate adaptation strategy. The NS forwards uplink to the appropriate AS and queues downlink from any AS to deliver the useful payload to the proper end-device. If a JS is present, Join-request and Join-accept messages are forwarded to the end-devices and the JS by the NS, respectively. The JS (which is described since release 1.1 of the specs) is in charge of managing end-devices affiliation in a secure, i.e. encrypted, way. Finally, being the application level protocol not described in the specifications, the AS has to implement it for delivering data to the final user.

Roaming is supported as well; accordingly, three roles actually exist for a NS; the only "serving NS" (sNS) controls the data link layer behavior of the end-device; the "home NS" (hNS) is actually connected to the AS; finally, several additional "forwarding NS" (fNS) can be connected to other gateways. If the sNS does not change, passive roaming is carried out; on the contrary, when handover roaming is enabled, the end-device is managed by the visited network, though user data are still forwarded to the original hNS.

It has to be highlighted that the protocols for implementing the interfaces between the fNS and the gateway, and between the hNS and the AS are not in the specifications. Only the communications among NSs and JS-NS interfaces is described. In the latter case, it is required the adoption of the HTTP protocol, encoding the payloads using JSON. As a matter of fact, several different proprietary implementations may exist. Such an architecture makes it easy

decoupling the owner of the infrastructure from the owner of the data, enabling new business scenarios.

For security purposes, the message payload is encrypted on a session base. End-device commissioning can occur in two ways; when over the air mode is used, the JS is involved for providing session keys and connecting the end-device with a hNS; otherwise, provisioned mode requires an out of band communication channel to provide the end-device with proper keys. Security on the backend is required as well, but implementation is left out to the implementer.

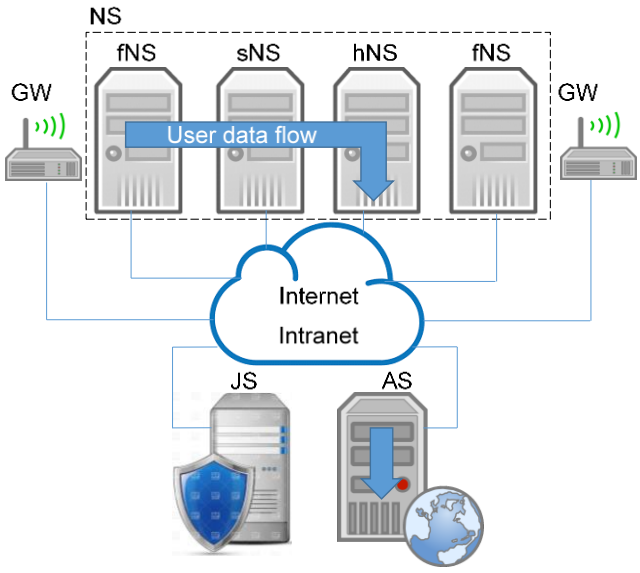


***Figure 2-2**: The LoRaWAN Architecture*

2.2.1.2.2 The proposed methodology for evaluating LoRaWAN networks

It is well known from the literature that the very simple wireless protocol adopted by LoRaWAN (exploiting reduced bandwidth, limited number of channels and ALOHA medium access mechanism with duty-cycle limitations) bounds end-to-end performance when very "dense" applications, in terms of both number of nodes and update interval needs, are considered [2.7]-[2.15]. On the other hand, from the previous discussion, it is evident that most of the complexity in a LoRaWAN resides in the backend; consequently, overall network performance is not only affected by the wireless tier, but by the backend as well. However, very few, if any, works exist in literature assessing the backend performance. Additionally, the loose description provided by the specifications (that only identifies the need of an Internet or Intranet as the backbone), allows for very different measurement setup that can make comparison a cumbersome task.

In this section, it was provided a general methodology for evaluating not only the wireless tier, but also the backend tier. In particular, the previously described network entities (i.e. the servers) are considered as black boxes and the focus is pointed on the actual message exchange. In this way, the proposed approach does not need architecture modification or additional code probes; moreover, it is not limited by the details (e.g. the number of wireless nodes or the deployment of servers in the cloud) of the implementation under test.

As asserted before, uplink is of main concern for typical LoRaWAN networks and the focus of the test methodology is about uplink data flow characterization. In this case, end-user traffic is injected in the LoRaWAN network under test by means of a probing node and it is collected out of the AS in accordance with the actual AS implementation. A simplified diagram is shown in **Figure 2-3**, where for sake of simplicity a single NS implementing all the network level functionalities is depicted; for the same reason, the (optional) JS, involved only in the key management, is not shown. In particular, in agreement with the LoRaWAN specifications, the uplink user data flow can be further split into the following sub-flows (it has to be noted that a similar approach can be used for evaluating the joining procedure as well):

- F0: from the end-user data source to the mote (data injection); it is out of the backend scope;
- F1: from the mote to the gateway; it is the only wireless portion of the overall data flow;
- F2: from the gateway to the NS;F3: from NS to the AS;
- F4: from the AS to the end-user data destination (data extraction).

Obviously, in order to evaluate time performance of implementation under test, time information must be added to each data flow; in turn the method to assign timestamps to events depends on the way the actual implementation is physically realized.

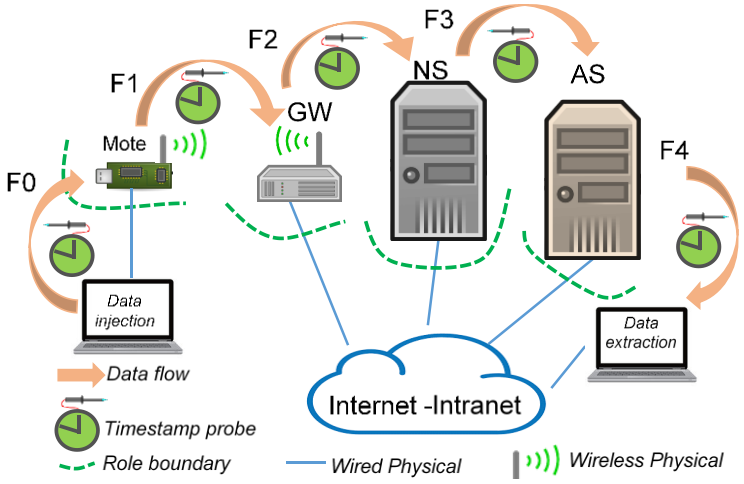


Figure 2-3: *The simplified diagram of the proposed methodology.*

As a consequence, the proposed methodology has three main steps (two preparation steps and one execution step):

1) Identification and block mapping: the real-world devices and software must be mapped on the corresponding backend entities described in the specification in order to identify boundaries between them and to highlight the ingress and egress ports of the information.

2) Timestamping point configuration: once boundaries between functionalities have been found, the position and the method to collect the time at which the information crosses the boundary (timestamping) is determined. For each identified sub-flow, up to two timestamps can be taken (origin timestamp and arrival timestamp). Many recording and timestamping techniques may be needed, depending on the physical nature of the information flow (inter-process exchange, network exchange, wireless exchange, cloud exchange). Since backend tier implementation can consist of several different physical devices, possibly spread on a large geographical area, physical layer adaptation and time synchronization to guarantee temporal coherence between timestamping probes may be needed. However, this can be generally achieved, since most of commercial solutions rely on office-like connectivity leveraging on standard Ethernet (IEEE802.3) and/or WiFi (IEEE802.11) links.

3) Measurements and analysis: the network under test is stimulated, and the information flow across it is monitored and measured. For instance, data are injected in a mote so that messages can be tracked and timestamped at all the timestamping points decided in the previous step. The final analysis is carried out calculating the architectural delays as differences between the timestamps relative to the same data flow. Many metrics can be defined depending on the focus of the analysis; with the proposed methodology, both communication delays through architecture blocks and communication delays due to network could be isolated and measured.

2.2.1.3 Application of the proposed methodology: two Use Cases

Currently, different implementations of LoRaWAN infrastructure exist; some of them are based on private backend servers and proprietary nodes that implements specific services [2.35]-[2.37], others, instead, offer free access for non-profit usage to an open-to-the-public infrastructure (but they pose some limitations to the available bandwidth) [2.38],[2.39]. Although the generality of proposed methodology makes it suitable for very large and complex real-world deployments, creating Use Cases that involve such big infrastructure is not in the scope of this section. In order to demonstrate the applicability of the proposed methodology,

without losing generality, this research considers two freely available solutions that are also used as basis for many LoRaWAN networks.

Note that the two implementations used in this section have not been optimized nor configured. They are simple reference implementations and this section uses them just to demonstrate how to highlight differences between systems under tests.

2.2.1.3.1 Use Case 1: The Semtech reference implementation

The Semtech corporation invented the LoRa radio and also proposed the first implementation of the LoRAWAN stack in order to promote its LPWAN transceivers. For instance, a reference kit is available, allowing easily setup of small, private LoRaWAN networks for demonstration purposes [2.40], while the document "LoRaWAN Network Server Demonstration: High Level Description" [2.41] provides an overview of the reference architecture. Interesting to note, the Semtech packet forwarder quickly spread among developers, thus becoming a de facto standard solution, adopted in both commercial and academic pilot tests [2.42][2.43]. All the installation and configuration manuals are made for developers that are willing to use Semtech devices to build end-nodes and gateway. Not all the source code or the datasheets are publicly available, some of them are released by Semtech after a Non-Disclosure Agreement is signed. The code of this "reference implementation" has been ported on several platforms, including Javascript, without any relation with Semtech official releases. As a consequence, the probability that given a LoRaWAN device is using some parts of the original work done by Semtech, is quite high.

The application of the first step of the methodology to the Semtech reference implementation reveals that it is strictly stuck to the LoRaWAN specification. The "actors" in this implementation are: the LoRaWAN end-device(s), the LoRaWAN gateway(s), the Network Server, and the Application server. The interactions within blocks are well defined and follow the path described in the specifications. Additionally, a Customer Server (CS) is defined as well, for allowing the end-user to access its own data.

There is direct communication between adjacent blocks; the coding scheme of this communication is defined in [2.44] and it is based on JSON objects encapsulated into IP/UDP or IP/TCP-based proprietary protocols. **Figure 2-4** shows the block diagram of the Semtech reference implementation and the path of an uplink message containing customer data.

The second step of the methodology requires the identification of the boundaries and the determination of the timestamping points for the exact calculation of the architectural delays. As discussed earlier, up to two timestamps can be assigned to each sub-flow. Thanks to the

plain structure of the Semtech implementation, the placements of the timestamping points is straightforward (see **Figure 2-4**):

- The data flow F0, out of the backend scope, is timestamped at T0 when the end user sends a message to trigger the LoRaWAN message transmission by the probing mote.
- The data flow F1, representing the wireless link, is timestamped at T1 capturing the over-the-air actual LoRaWAN user message; e.g. T1 could be recorded by the gateway when the packet arrives at its antenna, since the variation of the propagation time in air is negligible and nodes are located at known fixed position and propagation time can be compensated. Note that T1 is not collected in the considered experiments.
- The data flow F2 leaves the gateway at T2 and enters the NS at T3. The timestamps are taken looking at the network message containing the information about the received packet (coded with Semtech UDP/IP JSON format, thus allowing tracking of an individual message) with destination the NS.
- The data flow F3 leaves the NS at T4 and enters the AS at T5. The timestamps are taken looking at the network message containing the information (coded with Semtech TCP/IP JSON format, thus allowing tracking of an individual message) with destination the AS.
- The data flow F4 exits from the AS at T6 and arrives at the customer server at T7. Again, the timestamps are taken looking at the network message containing the information (coded with Semtech TCP/IP JSON format, thus allowing tracking of an individual message).

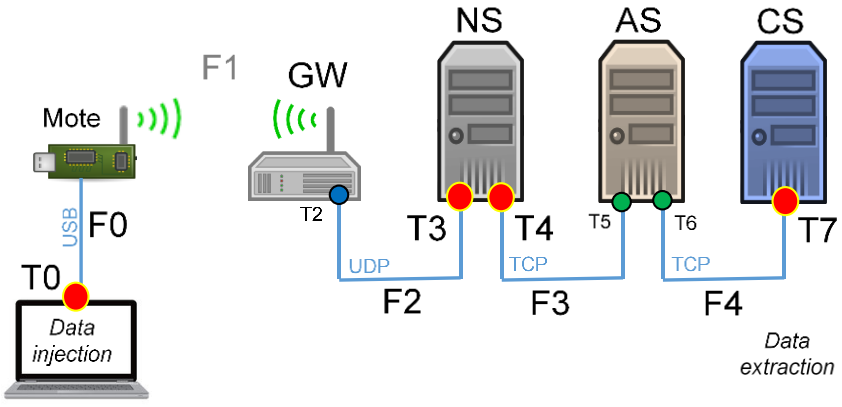


Figure 2-4: *The Semtech Architecture*

2.2.1.3.2 Use Case 2: The LoRa Server open source implementation

The LoRa Server is a LoRaWAN implementation which provides open-source components for building LoRaWAN networks. It is a ready-to-use solution that requires a low effort to have the system up and running. It includes a user-friendly web-interface, and REST API interface for integration with third party applications [2.46]. This implementation is used in several scientific works including [2.44].

The first step of the methodology was applied also to the LoRa Server implementation, highlighting some major architecture changes with respect to the LoRaWAN architecture as described in the specification. This does not mean that the LoRa Server implementation is not compliant with LoRaWAN; simply, the functionalities are spread in their own way in the system, or implemented with different techniques. From the documentation, the actors in this implementation are: the LoRaWAN end-device(s), the LoRa gateway (and the LoRa Gateway Bridge), the LoRa Server, the LoRa App Server, the Application and MQTT Broker.

The main innovation of this architecture, shown in **Figure 2-5**, is the use of a central MQTT Broker that acts as a central point through which all the information exchanges pass. This allows to make the system modular, separating the various services in a better way. Differently from the Semtech reference implementation, the LoRa Server gateways include a packet forwarder exchanging MQTT messages with the Broker. In this way, it is required only the knowledge about the corresponding MQTT topic in order to talk with the gateway (simpler and flexible configuration). Interesting to note, the LoRa Gateway Bridge has been proposed as well, i.e. a (software) component that transforms the IP/UDP protocol of the Semtech packet forwarder into messages over MQTT for the MQTT Broker. The LoRa Server component has the functionalities of the LoRaWAN Network Server component, while the LoRa App Server component has the functionalities of the LoRaWAN Application Server. Last, the Application is intended as the customer application that is using the end-device data.

In the second step of the proposed methodology, the identification of the boundaries and the determination of the timestamping points is done. The placement of the timestamping points (up to two for each sub-flow) is shown in **Figure 2-5**, and listed in the following:

- The data flows F0 and F1 are timestamped the same way of the Semtech reference implementation.
- The data flow F2 leaves the gateway at T2 and enters the NS at T3 as for the Semtech reference implementation. However, actual data exchange occurs via the MQTT Broker, but its presence is transparent from the LoRaWAN backend point of view and the related processing time is included in the F2 data flow. Additionally, if the LoRa Gateway Bridge is considered, an intermediate step occurs. The timestamps T2 and

T2b, taken looking at the UDP/IP transaction, refer to the egress time from the gateway and the ingress into the LoRa Gateway Bridge, respectively. LoRa Gateway Bridge egress timestamps is T2bm. The presence of the MQTT Broker allows to further identify timestamps T2mb, T2m and T3m.

- The data flow F3 leaves the LoRa Server at T4 and enters the AS (called LoRa App Server) at T5. The timestamps are taken considering the TCP messages exchanged by servers.
- The data flow F4 leaves the LoRa App Server at T6 and arrives at the Application at T7. Depending on the selected type of integration (leveraging the MQTT Broker or the REST interface), timestamping refers to the HTTP POST message or MQTT Publish message (T7m)

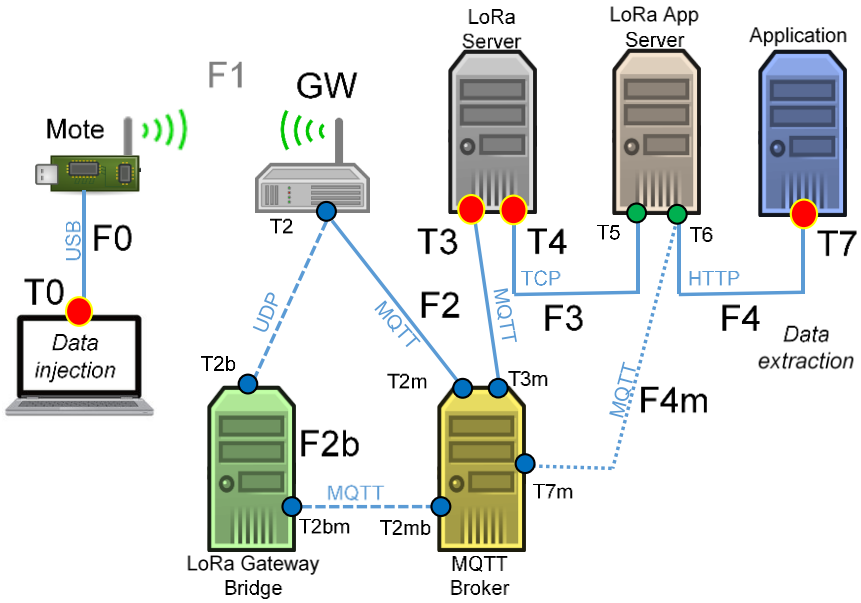


Figure 2-5: *The LoRaIO Architecture*

2.2.1.4 Experimental Setup and Results

Real-world experiments on the two previously described Use Cases have been carried out using a local network infrastructure (implemented by means of an Ethernet 100BaseT switch, as shown in **Figure 2-6**), without generality loss; the same experiments could be carried out if the backend was installed in the cloud. In particular, two different configurations have been considered: in the former, representing a possible small private LoRaWAN network, the complete backend is executed on a single machine; in the latter, the backend components are spread over two different machine, mimicking a larger public network where the NS is managed by the infrastructure owner, while the AS is handled by a different data/service

provider. Both the machines are equal; they host 4GB of main memory and rely on a AMD Athlon 64X2 5600 @ 2.9GHz. Regarding the operating system, it is different for the two considered Use Cases: the machines run Windows 7 Professional for the Semtech reference implementation and Linux Debian 9 for the LoRa Server open source implementation.

It has to be highlighted that the aim of the experiments is to verify the feasibility and flexibility of the proposed methodology. The considered configurations are not exhaustive, but represents possible real-world scenarios. The testbeds are not designed to stress the network infrastructure and evaluate performance in extreme conditions, but are intended to confirm the applicability of the proposed approach in real, possible very different, deployments.

More in detail, the LoRaWAN traffic is injected via a Microchip RN2483 LoRaWAN mote hosting a Semtech SX1276 transceiver and a Microchip microcontroller PIC18LF46K22. The RN2483 is a LoRaWAN 1.0 compliant modem that supports a virtual serial link over USB. The whole end device communication stack in managed by the modem, which, once properly configured, indefinitely listen for incoming user data to be transmitted. For sake of completeness, 18 B and 113 B long messages are sent by the mote, including the 13 B long LoRaWAN header. In both cases, messages are transmitted every 30 s using BW = 125 kHz, SF = 7 and CR = 4/5, on one of the default channel (centered around 868.3 MHz), having a duty cycle limitation d = 1% (according to EU limitations).

The LoRaWAN gateway is implemented by a Sentrius RG186; it hosts a pair of Semtech SX1257 analog frontend chips and the SX1301 baseband chip. The radio section is managed by a Linux-based WB50NBT module leveraging on an Atmel Cortex A5 @ 536MHz. It natively supports different packet forwarders, including those compatible with the Semtech and the Lora Server implementation. Additionally, only for Use Case 2, we performed experiments installing a LoRa Gateway Bridge on the same machine running the LoRa Server.

Timestamps have been collected using the Wireshark protocol analyzer, which uses time information from the local machine operating system kernel. The USBPcap software in Windows and USBmon in Linux are used for monitoring the USB link to the LoRaWAN mote. Regarding the Ethernet network connection, when the one-machine scenario is considered, timestamps refer to the internal loopback interface (RawCap software is installed for the Windows machine). On the contrary, when two-machine configuration is considered, timestamps refer to the network interface card.

It has to be highlighted that in the one-machine configuration, time synchronization is not an issue since all the timestamps natively refer to the same clock reference. On the contrary,

the two-machine configuration is an example of a distributed system and an adequate synchronization strategy must be considered. Two possible scenarios have been identified:

1. the Wireshark is installed in one machine only and all the timestamps refer to the entering/exiting traffic, mimicking the acquisition strategy of the one-machine configuration.
2. NTP synchronization strategy is exploited and each machine has its own Wireshark instance. In the considered scenario the NTP is installed in the local network at a single hop distance from the two-machines (**Figure 2-6**). The quality of the synchronization has been evaluated considering the residual offset estimated by the NTP daemon [2.47], [2.48]. During the experiments, in the worst case the mean offset value with the NTP time server was 0.4 ms, with a standard deviation of 0.2 ms.

An overview of the different setup configurations considered in these experiments are provided in Table 2-1.

Table 2-1: *Hardware and Software configurations for the experiments regarding Use Case 1 and Use Case 2.*

Configuration		**One-machine**	**Two-Machine**		**Two-Machines + NTP**	
Hardware Setup (both Use Case 1 and Use Case 2)		Mote	Mote		Mote	
		GW	GW		GW	
		Switch	Switch		Switch	
		-	-		NTP time server	
Software setup	**Use Case 1**	*Machine A*	*Machine A*	*Machine B*	*Machine A*	*Machine B*
		NS	NS	-	NS	-
		AS	-	AS	-	AS
		CS	CS	-	CS	-
		Wireshark	Wireshark	-	Wireshark	Wireshark
		USBpcap	USBpcap	-	USBpcap	-
		RawCap	-	-	-	-
		-	-	-	NTP client	NTP client
	Use Case 2	*Machine A*	*Machine A*	*Machine B*	*Machine A*	*Machine B*
		Lora Server	Lora Server	-	Lora Server	-
		LoRa App Server	-	LoRa App Server	-	LoRa App Server
		Application	Application	-	Application	-
		LoRa Gateway Bridge	LoRa Gateway Bridge	-	LoRa Gateway Bridge	-
		MQTT Broker	MQTT Broker	-	MQTT Broker	-
		Wireshark	Wireshark	-	Wireshark	Wireshark
		USBmon	USBmon	-	USBmon	-
		-	-	-	NTP client	NTP client

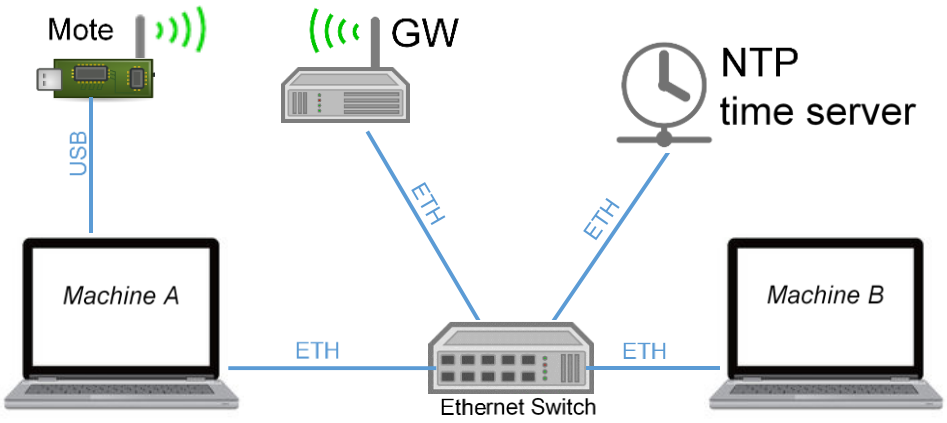


***Figure 2-6**: Hardware architecture for the experiments.*

2.2.1.4.1 Definition of LoRaWAN metrics

As previously stated, both the single and the two-machine configurations allow the acquisition of T0, T3, T4 and T7 timestamps, that are highlighted in RED in **Figure 2-4** and **Figure 2-5**. In this way, four metrics can be evaluated:

- D_{IN} = T3-T0, related to the time needed by a new uplink data to move along the F0, F1 and F2 data flows (including the mote and the gateway processing time);
- D_{NS} = T4-T3, related to the time needed by the NS to process the end-device data;
- D_{BE} = T7-T3, related to the time needed by a new uplink data to traverse the backend moving along the F3 and F4 data flows (including the NS and AS processing times);
- D_{LW} = T7-T0, related to the time needed by the data to traverse the entire LoRaWAN system.

Additionally, for the two-machine configuration, it is possible to consider timestamps T5 and T6 as well (highlighted in GREEN in **Figure 2-4** and **Figure 2-5**), thus allowing the computation of the additional metric:

- D_{AS} = T6-T5, related to the time needed by the AS to process the end-device data.

2.2.1.4.2 Evaluation of LoRaWAN metrics

The evaluation has been carried out estimating the previously described metrics over 200 experiments. In the **Figure 2-7** and **Figure 2-8**, boxplots are used for resuming results of the one-machine configuration for the Use Case 1 (Semtech reference implementation) and the Use Case 2 (LoRa Server implementation), when the dimension of the data varies from 100 bytes to 5 bytes. In the **Figure 2-9** and **Figure 2-10**, the results related to the two-machine configuration are proposed.

Thanks to the proposed methodology, it is possible to highlight that the metric D_{IN} is almost the same in both Use Cases, and it depends primarily on the dimension of the data to be exchanged. This result was expected since the hardware for the wireless transmission and reception is the same in the two implementations. It has to be highlighted that the use of MQTT for exchanging data from gateway and Lora Network Server in the Use Case 2 implementation does not affect the D_{IN} metric.

The general performance of the two Use Cases slightly decreases when two-machine configuration is considered because of the extra time (and grater variability) introduced by the network connection between the machines.

A more detailed overview of the performance of the two Use Cases is available in Table 2-2 and Table 2-3**,** where average and standard deviation values of the considered metrics are reported. It is evident that the Use Case 2 implementation seems faster in the conditions set by the experiments; however, these results are obtained under specific constrains in order to show the proposed methodology at work.

For the considered testbed, in which there is no contention for the medium access, the greatest part of the D_{LW} is due to the backend time D_{BE}. In particular, in the use cases the Application is simple and the most time consuming entity seems to be the Network Server (D_{NS}). The D_{AS} is 10 to 20 times faster than D_{NS}, as shown in Table 2-3 for the two-machine configuration synchronized with NTP

The proposed methodology allows to highlight the diverse behaviors imputable to the very different implementations of the two considered use cases: they have important architectural differences; they run under different operating systems, i.e., Windows for Use Case 1 and Linux for Use Case 2; and they use different network socket libraries. For instance, the performance of Use Case 1 worsens passing from one-machine to two-machine configuration, whereas this does not happen for Use Case 2. Hence the performance of Use Case 1 implementation is deeply affected by the use of network sockets between NS machine and AS machine.

The proposed methodology allows to easily evaluate the impact of communication protocol stack as well; indeed, the interval D_{BE}-D_{NS} is somehow related to the AS activity and can be compared with respect to the D_{AS} (when available). As previously stated, the time spent for accessing the network is included moving from one-machine to two-machine scenario. Looking at Table 2-2 and Table 2-3 it is evident that this access time is particularly significant for Use Case 1 (in the order of 80 ms), whereas it is reduced for Use Case 2 (less than 1 ms).

Additionally, the proposed methodology allows to highlight how peculiarities of backend implementation affect time-related performance. For instance, it is possible to study how NS implementation handles the multiple messages generated by co-located gateways that are listening at the same wireless frame from a single mote. The D_{NS} is insensitive to the message length but the NS dispatches the actual message only after a timeout expiration (sometimes addressed as the *deduplication delay*). The deduplication delay must be long enough to collect as many duplicates as possible, but short enough to give the AS the opportunity to reply to a message in time (for LoRaWAN Class A device, the first receive window may be opened just 1 s after the uplink transmission!). The experimental results are in agreement with the deduplication delay settings reported in both Semtech and LoRa Server documentation and satisfy typical value suggested by service providers (e.g. The Things Network operator reports an average delay lower than 100 ms [2.49]), and further validate the proposed methodology.

Analyzing the metric standard deviation is useful as well, since it provides some insights about the server(s) implementation and scheduling policy. For instance, large deviation is typically imputable to multimodal distribution emerging from task scheduling periodicity.

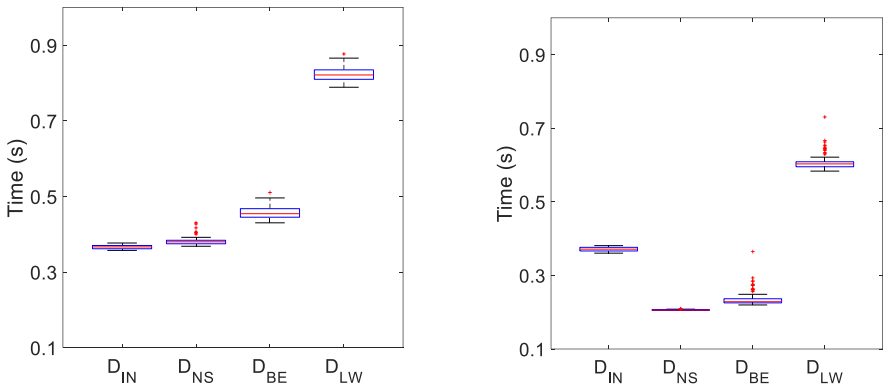


Figure 2-7: *Boxplots of the considered metrics for Use Case 1 (left) and Use Case 2 (architecture with full MQTT approach) (right) when the one-machine configuration is considered and an information of 100 bytes is traversing t*

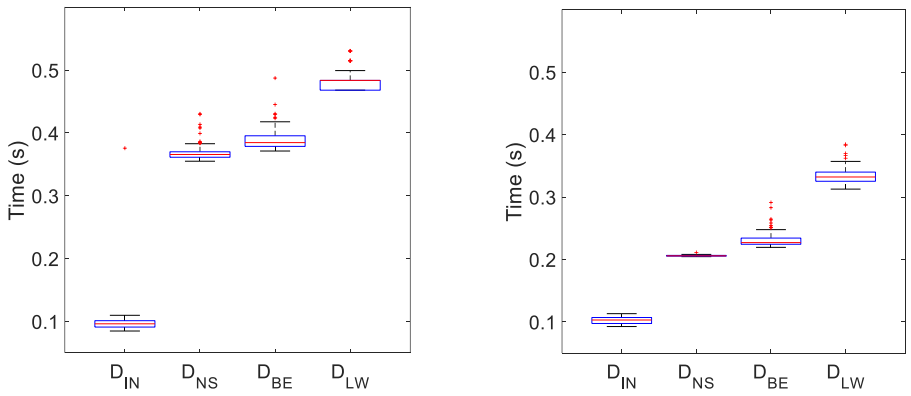


Figure 2-8: *Boxplots of the considered metrics for Use Case 1 (left) and Use Case 2 (architecture with full MQTT approach) (right) when the one-machine*

configuration is considered and an information of 5 bytes is traversing the LoRaWAN network.

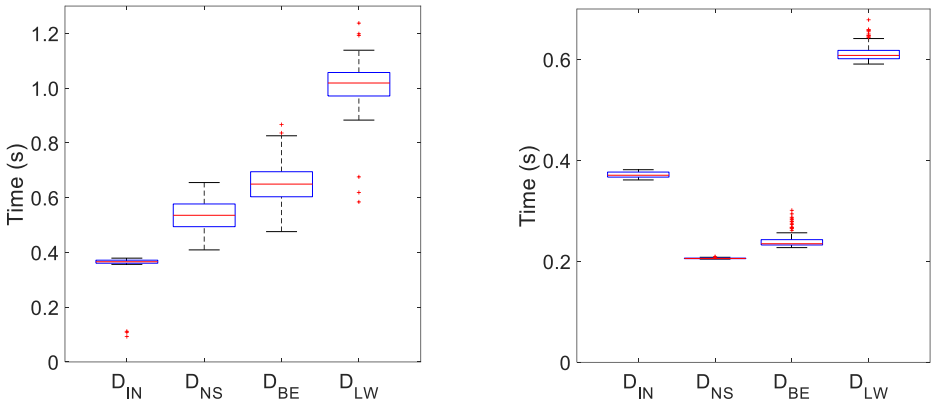


Figure 2-9: *Boxplots of the considered metrics for Use Case 1 (left) and Use Case 2 (architecture with full MQTT approach) (right) when the two-machine configuration is considered and an information of 100 bytes is traversing the LoRaWAN network.*

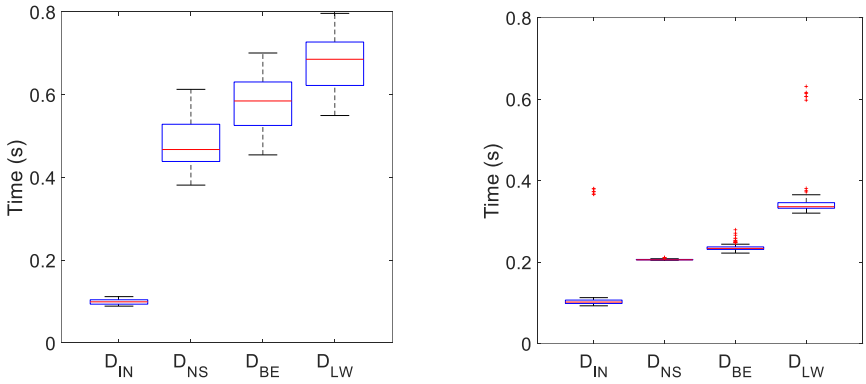


Figure 2-10: *Boxplots of the considered metrics for Use Case 1 (left) and Use Case 2 (architecture with full MQTT approach) (right) when the two-machine configuration is considered and an information of 5 bytes is traversing the LoRaWAN network.*

Table 2-2*: One-machine configuration: average and standard deviation values of the considered metrics for the Use Case 1 and the Use Case 2 (architecture with full MQTT approach).*

One-Machine Configuration									
Use Case 1					**Use Case 2**				
Metric	**5 bytes**		**100 bytes**		**Metric**	**5 bytes**		**100 bytes**	
	AVG [ms]	**STD [ms]**	**AVG [ms]**	**STD [ms]**		**AVG [ms]**	**STD [ms]**	**AVG [ms]**	**STD [ms]**
D_{IN}	95.9	7	366.5	5	**D_{IN}**	102.8	6	371.6	6
D_{NS}	369.7	15	381.7	11	**D_{NS}**	205.9	1	205.9	1
D_{BE}	390.8	21	457.2	16	**D_{BE}**	231.5	13	236.1	21
D_{LW}	486.7	23	823.8	18	**D_{LW}**	334.5	13	607.4	21

Table 2-3: *Two-machine configuration: average and standard deviation values of the considered metrics for the Use Case 1 and the Use Case 2 (architecture with full MQTT approach). The D_{AS} metrics is available only when the machines are synchronized with NTP.*

Two-Machine Configuration									
Use Case 1					**Use Case 2**				
Metric	**5 bytes**		**100 bytes**		**Metric**	**5 bytes**		**100 bytes**	
	AVG [ms]	**STD [ms]**	**AVG [ms]**	**STD [ms]**		**AVG [ms]**	**STD [ms]**	**AVG [ms]**	**STD [ms]**
D_{IN}	99.3	6	366.4	6	**D_{IN}**	102.1	5	371.3	6
D_{NS}	478.5	59	539.9	53	**D_{NS}**	206.1	1	205.9	1
D_{AS}	20.4	8	75.5	13	**D_{AS}**	29.0	21	34.6	21
D_{BE}	577.2	64	654.0	67	**D_{BE}**	236.5	9	241.8	16
D_{LW}	676.6	63	1020.4	66	**D_{LW}**	338.7	11	613.2	17

2.2.1.4.3 Implementation related metrics and their evaluation

The previously analyzed metrics can be evaluated on any LoRaWAN implementation, since they rely on mandatory network components, whose behavior is described in the specifications. However, depending on the peculiar implementations, some ancillary components may exist as well and additional metrics can be consequently defined. For example, the Use Case 2 includes the LoRa Gateway Bridge and the MQTT Broker; the proposed approach can be easily further extended including the following metrics:

- D_{GB} = T2bm-T2b, it is related to the time needed to traverse the LoRa Gateway Bridge along the F2b sub flow (see **Figure 2-5**);
- D_{BR} = T3m-T2m, related to the time needed by the MQTT Broker to receive and distribute to subscribers a new message.

As highlighted by results shown in Table 2-4, the impact of both the LoRa Gateway Bridge and the MQTT Broker is negligible in the considered simple testbed.

Table 2-4: *Use case 2 related metrics: time needed to traverse the LoRa Gateway Bridge (D_{GB}) and time related to the MQTT Broker processing a new message (D_{BR}).*

Use Case 2 – impact of specific architecture parts									
One-Machine Configuration					**Two-Machine Configuration**				
Metric	**5 bytes**		**100 bytes**		**Metric**	**5 bytes**		**100 bytes**	
	AVG [ms]	**STD [ms]**	**AVG [ms]**	**STD [ms]**		**AVG [ms]**	**STD [ms]**	**AVG [ms]**	**STD [ms]**
D_{GB}	0.4	0.2	0.4	0.1	**D_{GB}**	0.4	0.1	0.4	0.1
D_{BR}	0.7	0.4	0.6	0.4	**D_{BR}**	0.3	0.3	0.6	0.4

Table 2-5 resumes results of Use Case 2 experiments when the LoRa Gateway Bridge is executed on the same machine running the LoRa Server. These results can be compared with Table 2-2 and Table 2-3. As expected, D_{IN} is smaller, since the LoRa Gateway Bridge is executed by a PC-class machine and a simpler packet forwarder (without MQTT socket support) is executed by the reduced capability microprocessor of the gateway. Additionally, despite the D_{NS} parameter has similar values in all scenarios involving Use Case 2, the D_{BE} for small packets changes depending if the LoRa Gateway Bridge is executed or not. Hence, the AS (i.e. the LoRa App Server) has a different execution time. As previously stated, the time spent for the AS activity can be estimated by means of the D_{BE}-D_{NS} quantity. The relative frequency histogram of the D_{BE}-D_{NS} quantity is plot in **Figure 2-11**, confirming the time interval distribution is wider when the LoRa Gateway Bridge is not executed.

Use Case 2 also offers different integration modes for the end-user application: based on MQTT Broker subscription or via HTTP message using the POST method. In Table 2-6, the two modalities are compared for the one-machine scenario. It can be highlighted that HTTP based integration shows larger delays, probably due again to task scheduling, as can be inferred from the analysis of the D_{BE}-D_{NS} time interval. For instance, in the case of messages with 5 bytes information and HTTP integration, **Figure 2-11** shows a bimodal distribution, usually related to software timeouts.

Table 2-5: *Use Case 2 with LoRa Gateway Bridge: average and standard deviation values of the considered metrics with one-machine and two-machine configuration.*

Use Case 2 – with LoRa Gateway Bridge									
One-Machine Configuration					**Two-Machine Configuration**				
Metric	**5 bytes**		**100 bytes**		**Metric**	**5 bytes**		**100 bytes**	
	AVG [ms]	**STD [ms]**	**AVG [ms]**	**STD [ms]**		**AVG [ms]**	**STD [ms]**	**AVG [ms]**	**STD [ms]**
D_{IN}	99.7	6	367.6	6	**D_{IN}**	99	6	367.1	6
D_{NS}	205.8	1	205.7	1	**D_{NS}**	205.5	1	206.0	1
D_{BE}	229. 9	11	235.9	16	**D_{BE}**	228.1	11	241.7	20
D_{LW}	330.7	13	604.7	17	**D_{LW}**	328.2	12	609.9	20

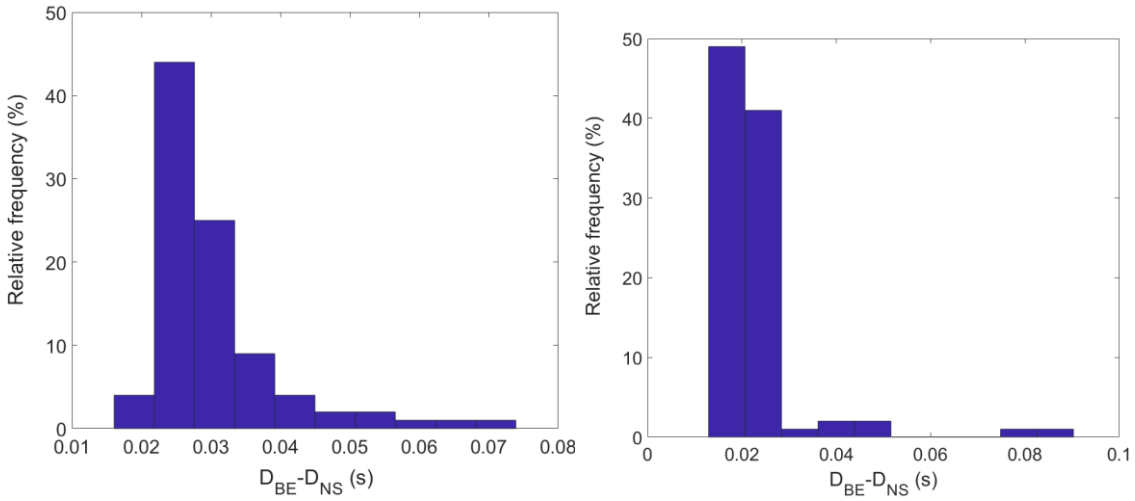


***Figure 2-11**: Relative frequency (in percentage) of the time interval DBE-DNS for Use Case 2 – two-machine configuration without the LoRa Gateway Bridge (left) or with the LoRa Gateway Bridge (right) and an information of 5 bytes is traversing the LoRaWAN network*

***Table 2-6:** Use Case 2 one-machine configuration without the LoRa Gateway Bridge or with the LoRa Gateway Bridge when the integration mode is MQTT or HTTP.*

	Use Case 2 – impact of Application interface (One-Machine)								
	without LoRa Gateway Bridge					**with LoRa Gateway Bridge**			
	Metric	**5 bytes**		**100 bytes**		**Metric**	**5 bytes**		**100 bytes**	
		AVG [ms]	**STD [ms]**	**AVG [ms]**	**STD [ms]**		**AVG [ms]**	**STD [ms]**	**AVG [ms]**	**STD [ms]**
MQTT	**D_{BE}**	231.5	13	236.1	21	**D_{BE}**	229. 9	11	235.9	16
	D_{LW}	334.5	13	607.4	21	**D_{LW}**	330.7	13	604.7	17
HTTP	**D_{BE}**	344.9	241	396.0	313	**D_{BE}**	422.4	344.9	356.2	254
	D_{LW}	447.8	241	767.3	313	**D_{LW}**	523.3	346.1	724.9	256

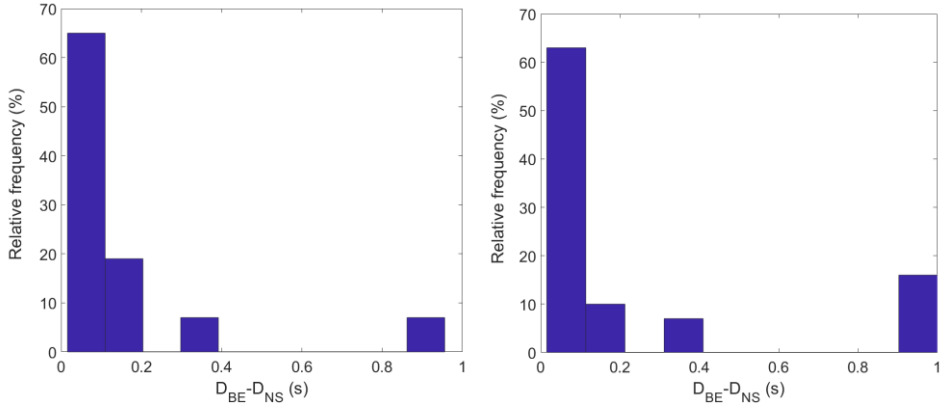


***Figure 2-12**: Relative frequency (in percentage) of the time interval DBE-DNS for Use Case 2 – one-machine configuration and HTTP integration mode without the LoRa Gateway Bridge (left) or with the LoRa Gateway Bridge (right) and an information of 5 bytes is trave*

2.2.1.5 Final Considerations for Architecture Delays in Private LoRaWAN networks

This research presents a methodology for the experimental assessment of architectural delays of LoRaWAN implementations. The new approach has three phases: a mapping phase that identifies the LoRaWAN blocks, as defined in the specification, inside the implementation under test; a setup phase in which the relevant timestamping points are isolated inside the architecture; and last, a measurement phase where the delays are evaluated along the information path from LoRaWAN node to LoRaWAN user application. The section is completed by two Use Cases that demonstrate the use and the effectiveness of the proposed methodology.

The experimental results clearly show that the proposed methodology is useful to compare the performance of entire LoRaWAN systems (Use Case 1 versus Use Case 2), with great benefit of the owner of the infrastructure and of the end user. The high level of details of the calculated metrics can be directly used to make implementation choices or to maximize results of optimization actions. Specific view of implementation particularities (e.g. for Use Case 2) makes the proposed methodology also suitable for architecture designers and stack developers.

2.2.2 Public LoRaWAN networks

The aim of this research is to further characterize real scenarios, experimentally verifying the overall delay experienced by several end users when they are located in possibly very large areas. In particular, the original contribution of the research is the implementation of a real world use case and the definition of some metrics that permit to formally estimate such a delay. The considered LoRaWAN infrastructure has been developed within the "Brescia Smart Living" project and is an example of a public LoRaWAN.

2.2.2.1 The "Brescia Smart Living Project"

The "Brescia Smart Living" (BSL) project, granted by the Italian Ministry of Education, Universities and Research, is aimed to develop and test IoT-related solutions in order to make smartcity of the future more efficient, comfortable and sustainable [2.51]. The project aims to improve the citizen quality of life, leveraging on all the disruptive technologies proposed for the smartcity management. In particular, all the aspects involved in the management process of medium-sized city have been considered:

- At the domestic level, the goal is to augment the user awareness about resource (energy, gas, water…) consumption and stimulate more conscious lifestyle;

- At the energy network level, the goal is to acquire data not only from the electric energy distribution grid (the so called smartgrid), but to integrate gas, water and district heating grids, public lighting, waste collection services etc. for better and more efficient control strategies;
- At the city and social level, the goal is to promote relationships among citizens and administration activities;
- At the street and district level, the goal is to develop a communication infrastructure supporting affordable, ubiquitous communication.

2.2.2.1.1 BSL and the LoRaWAN

Several communication technologies have been taken into account in the BSL pilot tests, including WiFi, mobile communications and LPWANs. In particular, LoRaWAN has been considered an effective solution for ubiquitous connectivity in most of non real-time applications. For instance, LoRaWAN is currently used for monitoring the filling level of smart bins and dumpsters and for accessing pressure and temperature data from the district heater network. In all these applications, the update time is limited to few readings per day (waste management) or few readings per hour (district heater management). The LoRaWAN infrastructure of the BSL project is based on the Patavina NetSuite solution from A2A Smart City (see the architecture diagram shown in **Figure 2-13**). It is a complete solution for the whole backend and permits end user application integration by means of an MQTT Broker. Indeed, each end node uplink is published by the Broker as an MQTT topic which can be subscribed by all the interested end users.

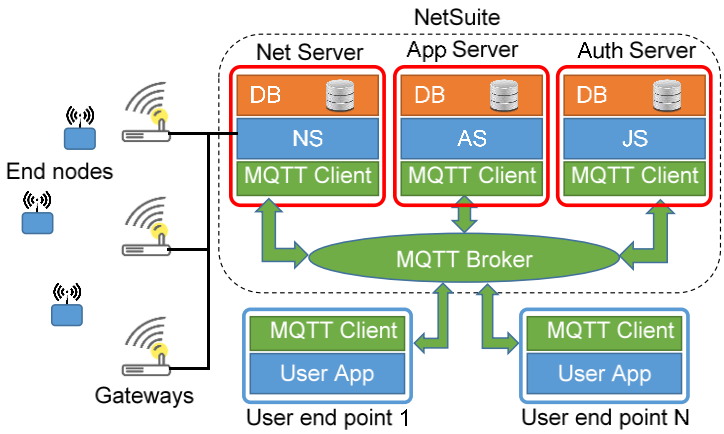


***Figure 2-13**: Architecture of the LoRaWAN solution used in the BSL project*

It has to be highlighted that the whole urban area of the city of Brescia is covered with LoRaWAN; in particular, the NetSuite infrastructure serves end nodes within an area of 80 km^2 using more than 100 LoRaWAN gateways.

2.2.2.2 The considered use case

The availability of an AS allows remote users to easily access data from end nodes. However, any communication system including a backend implemented in the Cloud, is typically considered as a "black box" for the user. In these cases, an effective way to estimate the overall latency is to design a specific experiments representing a typical use case. In this section, a formal methodology, based on well-defined metrics, to experimentally measure the end-to-end delay of a LoRaWAN application is described and applied to a purposely designed setup. Additionally, the proposed methodology can be further used to evaluate the capability of different end users, located in a possibly vast geographical area, to simultaneously identify the same event. As stated in the introduction, only few works exist dealing with the delays introduced by the overall communications infrastructure. Nevertheless, application level delays greatly influence the value of the offered service. Authors try to provide an insight about such delay using the test procedure described in the following.

2.2.2.2.1 Reference time distribution.

Starting from its definition, end-to-end delay can be evaluated as the time difference between the arrival time and the starting time of the information of interest. However, due to the distributed architecture of the considered scenario, a common reference time must be provided somehow to all the actors involved. In this section, the Coordinated Universal Time (UTC) has been adopted, since it can be easily derived from GPS (Global Positioning System) receivers or Internet via synchronization protocol like NTP (Network Transfer Protocol). Unfortunately, NTP uncertainty greatly depends on the time-related performance (e.g. the latency) of the Internet connection, and typically is estimated in tens of ms [2.48]. On the contrary, GPS receivers derive the reference time by the stable atomic oscillators on satellites and can offer offset errors well below one ms even with low-cost receivers.

In this section, data producer and consumers only have a network port, so that a direct connection with the GPS receiver reference signal (e.g. 1-Pulse Per Second) is not feasible. Consequently, (as shown in **Figure 2-14**) a commercial NTP server TM1000A (from TimeMachine) disciplined by GPS receiver has been considered for remote endpoints (via a point to point network connection on a dedicated Ethernet card), while direct connection to INRIM NTP server (Stratum 1) is used for both source and local endpoints. In all the cases, the network delays are minimized and only the variability due to the NTP implementation affects the synchronization accuracy. In particular, the quality of the time reference can be

derived from the very informative NTP statistics [2.45]. Fortunately, the LoRaWAN communication infrastructure implements its own time synchronization procedure; in particular, the Patavina NetSuite provides an UTC timestamp in each MQTT topic published after an uplink is received by the AS, which can be further used for delay estimation.

2.2.2.2.2 Description of the experimental procedure

The block diagram of the proposed experimental setup is shown in **Figure 2-15.** A single LoRaWAN end node is implemented by a PC (i3 processor) running a Linux Debian 9 distribution (kernel release 4.9) connected with a Microchip LoRaWAN dev kit (DM164138, hosting the modem RN2483), via an USB link. Four end nodes (EP1-EP4) have been implemented using the Siemens IOT2040 device (featuring the energy-saving Intel Quark x1020 (+Secure Boot), 1 GB RAM, 2 Ethernet ports, 2 x RS232/485 interfaces, battery-powered RTC); the operating system is Yocto Linux (kernel release 2.2.1).

The IOT2040 software is developed using the Node-RED environment to simplify the development process and to ensure portability on different hardware platform. EP1 and EP2 have been connected to the university network (high speed Internet access); EP3 is placed in the city of Brescia with a standard ADSL connection; EP4 is placed in Milano (70 km from Brescia) with a standard ADSL connection.

The PC executes the Wireshark protocol analyzer and collects the USB messages by means of the *usbmon* interface. In this way timestamp T1 is assigned when an LoRaWAN uplink transmission is started. Each EP_n is a MQTT subscriber of the topic "event of interest" in the Patavina NetSuite MQTT Broker; when a new message is received, the notification is timestamped as $T3_n$. Last, the timestamp T2 is taken when the "event of interest" is recognized by the AS. Given those timestamps, the following metrics are defined: the LoRaWAN backbone delay ND=T2-T1; the MQTT Broker delay $MD_n=T3_n-T2$; and the overall end-to-end delay $OD_n=T3_n-T1$. Last, the asymmetries between the OD_n of the various information endpoints EP_n can be calculated as $A_{xy}=OD_x-OD_y$.

The measurement uncertainty of the OD delay mainly depends on: synchronization uncertainty between nodes (u_{sn}), and event timestamping uncertainty (u_{tn}). The former is obtained from the residual offset values in NTP logs; the latter is evaluated analyzing timestamps marking the beginning (T_{start}) and the end (T_{stop}) of a fixed delay, whose duration is estimated as $\Delta = T_{stop}-T_{start}$. Since timestamping uncertainty u_{tn} affects both timestamps, $u_{tn}^2 = \frac{u_\Delta^2}{2}$. The whole procedure for the uncertainty estimation is reported in [2.45], where it is suggested that the average error μ and the standard deviation σ of the measurement populations

must be taken into account, since the systematic errors introduced by the setup cannot be estimated and compensated. For this reason, $u_{sn} = \sqrt{\mu_{sn}^2 + \sigma_{sn}^2}$ and $u_{tn} = \sqrt{\frac{\mu_{\Delta}^2 + \sigma_{\Delta}^2}{2}}$. The same characterization procedures have been applied in this section, leading to results in 0.

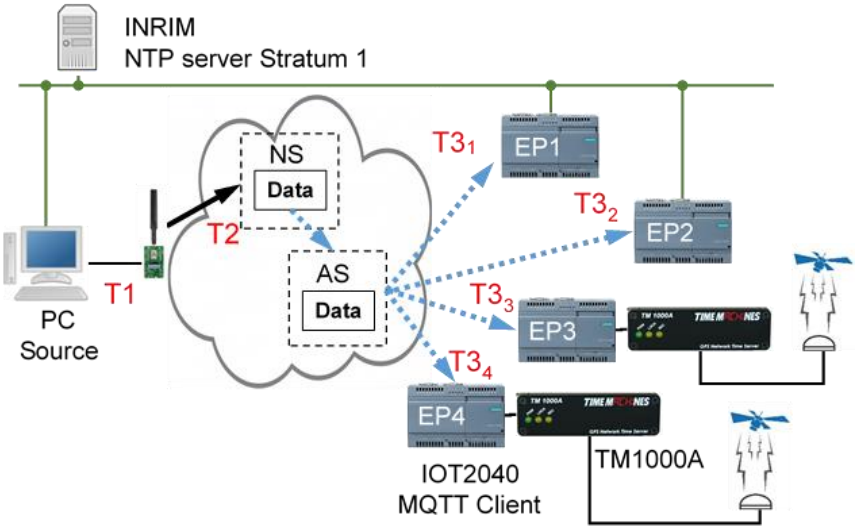


***Figure 2-14**: Block diagram of the proposed experimental setup*

The distributions of the time offset of the considered devices with respect to the UTC are shown in **Figure 2-16**; the synchronization error is always less than 1 ms in all the devices. On the other hand, the timestamp uncertainty of the considered platform (Source PC and endpoint IOT) has been derived respectively from [2.52] and [2.23], respectively.

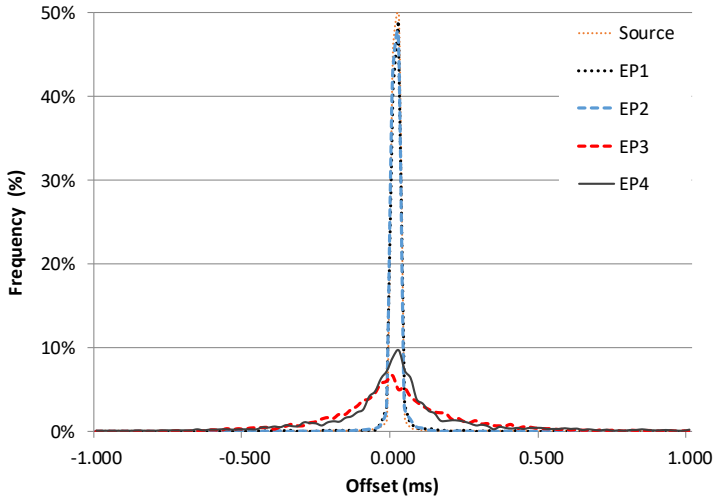


***Figure 2-15**: Distribution estimates of the NTP synchronization error.*

The overall uncertainty of the metrics defined above can be estimated as $u_{pq}^2 = u_{sp}^2 + u_{tp}^2 + u_{sq}^2 + u_{tq}^2$, where p and q are the two nodes that captured the timestamps used for the considered metrics. Note from Table 2-7 that u_{tn} is always dominating over the u_{sn}.

Table 2-7: Synchronization and timestamp uncertainty (MS)

	u_{sn}	μ_{sn}	σ_{sn}
Source	**0.018**	0.000	0.018
EP1	**0.027**	0.001	0.027
EP2	**0.026**	0.001	0.026
EP3	**0.234**	0.007	0.234
EP4	**0.209**	0.003	0.209
	u_{tn}	μ_{Δ}	σ_{Δ}
Source	**0.7**	1.0	0.1
EPX	**2.6**	3.1	2.1

2.2.2.2.3 Experimental results

The overall experiment duration is one day, with the collection of a total of 1440 messages transmitted every one minute. The length of the user message is 30 B and includes the transmission timestamp and a sequence number for sorting. Additional LoRa physical layer and the 5 B of data link headers and trailers are added, resulting in an overall time on air of about 226 ms (SF = 7 and CR = 4/5 are used).

The results and the distribution of the OD delay reported in Table 2-8 (and graphically in **Figure 2-16**), show that three endpoints are in good agreement since they present very similar outcomes with average delay of about 500 ms. The EP4, which is placed in the most remote location has higher variability, because of the poor Internet connection. However, a strange behavior has been immediately noticed in all the experiments: the presence of very few outliers (less than 50 in a day) with delay longer than 4 s. A deeper analysis of this result is possible from Table 2-9 and Table 2-10. The longer delays ND are generated inside the LoRaWAN backend, while the MQTT Broker delay MD is faster and constant (EP4 is again the worst).

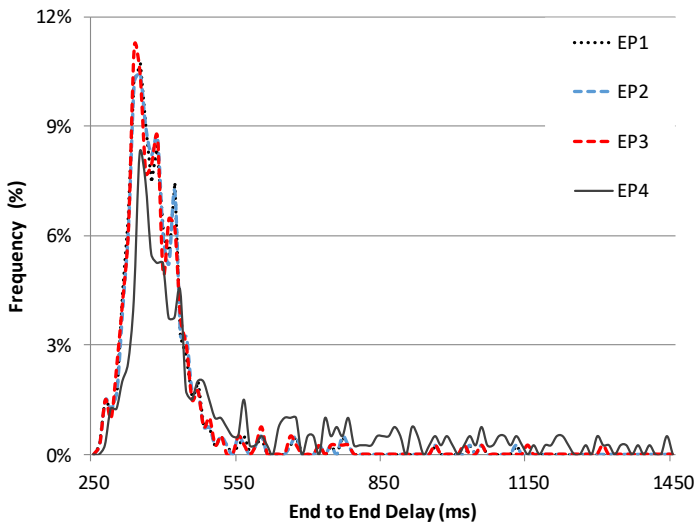


Figure 2-16: *Distribution estimate of the overall End-to-End delay OD*

The last evaluation regards the asymmetry between the end-to-end delay. This metric is important when events sensed in the field are used to trigger actions in a distributed environment: the simultaneity of distributed actions related to the same event is bounded by the asymmetry. **Table 2-11** demonstrates that it is possible to reach very low asymmetry values only when stable Internet connection is used.

Table 2-8: End to End delay OD of the different endpoints (MS)

	EP1	EP2	EP3	EP4
Max	4785	4785	4784	6577
95 perc	735	735	752	1504
Average	485	486	487	675
5 perc	300	300	298	311
Min	260	259	258	269
Std. Dev.	602	602	604	733

Table 2-9: MQTT broker delay MD (MS)

	EP1	EP2	EP3	EP4
Max	129	129	928	6252
95 perc	86	88	86	948
Average	44	45	47	234
5 perc	15	15	13	25
Min	14	14	12	21
Std. Dev.	22	22	51	451

Table 2-10: LoRaWAN backbone delay ND (MS)

Max	95 perc	Average	5 perc	Min	Std. Dev
4752	692	441	263	240	600

***Table 2-11**: Asymmetry between the Endpoints (MS)*

	A_{21}	A_{31}	A_{23}	A_{41}	A_{42}	A_{43}
Max	57	879	878	6202	6202	6204
95 perc	1	5	5	898	893	899
75 perc	0	-1	-1	222	221	223
50 perc	0	-2	-2	11	11	13
25 perc	0	-2	-2	8	8	10
5 perc	0	-3	-3	7	7	8
Min	-1	-7	-59	4	-15	-868
Aver.	1	2	2	190	189	188
Std.Dev.	5	46	46	448	448	452

2.2.2.3 Final Considerations for Architecture Delays in Public LoRaWAN network

This section proposes both a measurement method and the related metrics to formally evaluate the delays at the application level. The proof-of-concept is the implementation of a real use case based on a public LoRaWAN infrastructure. A purposely designed setup, synchronized by means of the NTP protocol, has been prepared to evaluate of the complete delay chain from the source of data (connected with LoRaWAN) and the final user of data (connected using Internet). The average overall end-to-end delay is on the range 400-700 ms, but peaks of several seconds have been recorded. The deeper analysis showed that the reason of such sporadic long delays is within the LoRaWAN backend, and not in the data distribution system based on MQTT protocols. On the other hand, low asymmetry values between endpoint delays are possible only when stable Internet connections are used; in such cases, the simultaneity of distributed actions related to the same event can be guaranteed within few milliseconds. Interesting to note, results agree with findings of previous work 2.50 and extend the characterization to a real multi-user scenario.

2.3 LoRaWAN applications in a Smart City context

This section devises the jointly use of LPWANs with widely-diffused and well-accepted localization techniques, as the Global Positioning Systems (GPS, outdoor) and real-time location systems (RTLS, indoor), for Smart Campus applications. In particular, a LoRaWAN node equipped with both GPS and Ultra Wide Band-based UWB-RTLS has been developed and tested in real-world scenarios. Experimental results demonstrate the feasibility of the proposed approach; in particular, location errors are in the order of few tens of meters for GPS and in the order of few tens of centimeters for UWB.

The main original contributions of this research are:

- the proposal of joint use of GPS (Global positioning System) and RTLS-UWB (real-time localization systems based on ultra-wide band wireless links) for complementing a LoRaWAN node in Smart Campus applications for providing position indoor and outdoor;
- evaluation of the obtainable performance in a real-world scenario by means of a purposely designed test bench, confirming the feasibility of the proposed approach.

2.3.1 Node localization

Availability of location data is fundamental for many Smart Campus services. Localization is typically classified as range-free and range-based; the former, that does not use specialized hardware for distance measurement, is allowable only if coarse localization is tolerated; consequently, range-based is generally preferred. Although the use of satellite-based solutions solved the problem of outdoor positioning, indoor localization remains an interesting research topic. Many technologies have been suggested in the past to deal with such a problem: Radio-Frequency Identification [2.53], Bluetoot [2.54], Wireless Local Area Network, i.e., Wi-Fi, [2.55]-[2.58], ZigBee [2.59], ultrasonic [2.60], magnetic [2.61], and even based on LED lighting [2.62], just to mention a few. The lesson learnt is that reliable and accurate results can be easily obtained using time-of-flight ranging. In this section, the joint adoption of well-accepted satellite-based outdoor localization and innovative UWB indoor localization is suggested. For the sake of completeness, some considerations about the two different methodologies are resumed in the following.

2.3.1.1 Outdoor localization

Outdoor localization is typically carried out by means of so called Global Navigation Satellite Systems (GNSSs), using the multilateration technique. Ranging is based on time of flight estimation, thus accuracy of time measurements in both satellites and on-the-field nodes is fundamental for minimizing the overall localization error. Satellite clocks are continuously checked by ground control stations; the time deviation from ideal behavior is computed and passed to the corresponding satellite for correction. Node clocks leverage on crystal oscillators, since only short-term stability is of main importance. Additional errors are due to the uncertainty in satellite position. Signal degradation due to interferences from other radio signals, reception from multiple paths, and atmospheric effects in the troposphere and ionosphere (which change the speed of the traveling wave) further worsen localization

accuracy. The location errors are on the order of few meters [2.63]. As a concluding remark, it must be remembered that GPS receivers need to be synchronized with satellites and require some time to compute the location (fix). Three different fix conditions exist: hot, warm, and cold, with time to fix ranging from less than 1 s to about 12 minutes (needed to acquire the so called navigation message).

2.3.1.2 Indoor localization

Although outdoor localization is generally (and easily) solved using one of the several GPS modules available on the market, indoor localization is a tricky task and an active research field [2.64]. Indoor localization is typically carried out by RTLS systems, which can address very different applications, as highlighted by the ISO/IEC 24730 standard. In range-based indoor localization systems exploiting time of flight estimation (the most accurate and versatile), the role of satellites in GNSS is assumed by nodes in fixed and well-known locations, the anchors. Despite optical and inductive links can be exploited as the physical layer, most of solutions rely on radio frequency links in unlicensed bands. The location methodology is generally multilateration, based on the measurement of the difference in distance of a node (the tag, according to the RTLS nomenclature) with respect to two or more anchors. In particular, the Symmetric Double Sided Two-Way Ranging (SDS-TWR) technique is preferred. In the SDS-TWR, the role of the initiator and the responder among the two involved nodes is exchanged, as shown in **Figure 2-17**. The initiator is in charge of starting the transaction sending a message and after a fixed amount of time Δt_{reply} the responder sends back another message. Time of flight t_p is estimated from time of arrival (ToA) of messages.

Differently from GNSS, requiring the satellite clocks to be strictly synchronized, SDS-TWR is an asynchronous procedure. As a consequence, requirements on tag clock are relaxed (since Δt_{reply} can be made short and repeatable), but accurate ToA estimation is crucial. For this reason, UWB communications based on very short pulses are considered an effective approach for RTLS, since they allow very accurate ToA estimation. For instance, using the IEEE802.15.4-UWB physical layer, sub-meter errors on the location are obtainable [2.65]. An example of such a RTLS is the DecaWave DWM1001, which is better described in the next section. Due to the very interesting nominal performance, it has been chosen for the Smart Campus experimental.

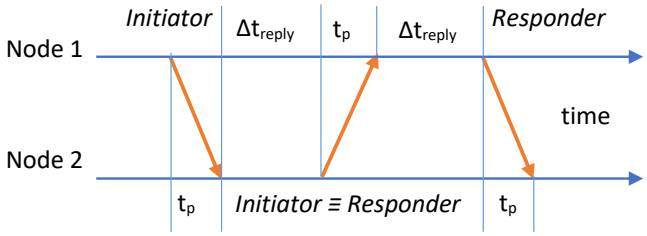


Figure 2-17: *Symmetrical double-sided two-way ranging (SDS-TWR).*

2.3.2 The infrastructure for indoor localization: the DecaWave RTLS solution

Differently from outdoor localization via GSNN, the end-user is directly involved in the deployment of an indoor RTLS infrastructure, as stated in the previous section. For instance, the DecaWave RTLS (DRTLS) solution adopted in this section includes the following components (as in **Figure 2-18**): fixed nodes (anchors), used as reference nodes for the location procedure (three to four of them are required for locating a tag); mobile nodes (tags), i.e., the nodes to be localized.

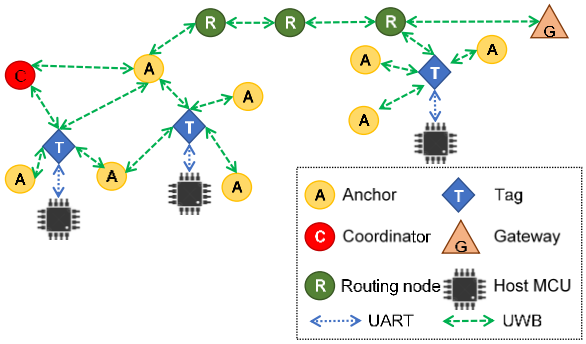


Figure 2-18: *The DWM1001 Real-Time Location System (DRTLS) network and its components.*

The core of each node is the DWM1000 UWB transceiver, compliant with the IEEE802.15.4 UWB physical layer and hosting a Nordic Semiconductor nRF52832 microcontroller unit (MCU) capable of Bluetooth Low Energy (BLE) connectivity. Indeed, BLE can be used for connecting external devices (e.g., PC/server and tablet/smartphone) for configuring the DRTLS nodes via an out-of-band link; another opportunity is the use of an UART wired link. In particular, in any DRTLS network one of the anchors must be configured as the DRTLS coordinator (C), in agreement with the IEEE802.15.4 specifications. The coordinator is in charge of starting, synchronizing and controlling the network; it advertises its presence via a Beacon Message, allowing other additional anchors to join the network.

The DRTLS physical layer uses impulse radio UWB (IR-UWB) and operates in the 6.5 GHz channel. Each symbol is modulated according to the burst position modulation with

binary phase shift keying (BPM-BPSK) with a pulse repetition rate of 64 MHz. The raw bit rate is about 6.8 Mbps. The DRTLS exploits a Time Division Multiple Access (TDMA) medium access policy, which in turn requires the nodes to be synchronized with a common clock. The data exchanges are organized in a repeating "superframe" structure, whose duration is 100 ms (see **Figure 2-19**). Anchors announce their position by means of Beacon Messages sent in the first sixteen slots (named BCN in **Figure 2-19**). BCN0 is occupied by the coordinator; additional anchors, once synchronized with the coordinator, can transmit Beacons using the other BCN slots. In other words, no more than sixteen active anchors can operate in the same area. Beacons are used by tags for synchronizing as well. Once the tag has estimated and compensated the time error of its local clock with respect to the network time, it can start to transmit the data in the assigned TWR slots. In particular, the SDS-TWR mechanism leverages on Poll (from anchor to tag), Response (from tag to anchor) and Final (from anchor to tag) messages, as for the IEEE802.15.4-UWB specifications. The Service slots are used for ancillary data exchange, as the network join requests or the list of anchors serving a tag. The concluding Idle time can be used to transmit additional backhaul data.

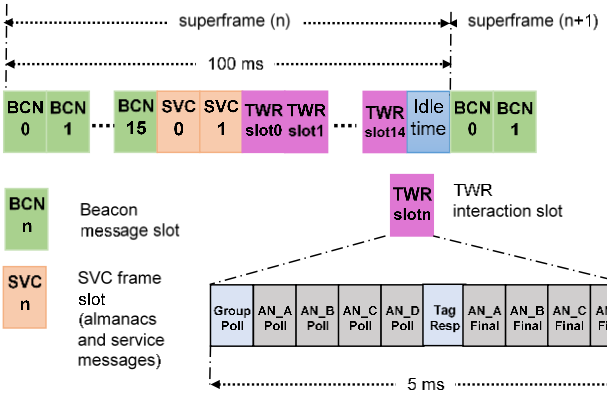


Figure 2-19: *DWM1001 Real-Time Location System (DRTLS) network and its components.*

.Regarding the network architecture, multi-hop topology is permitted for propagating data on a wider area. However, due to the superframe arrangement, some scaling rules must be taken into account. In particular, the maximum system capacity is 150 node/s, that is 15 tags can be located every 100 ms (the superframe length), or 150 tags every second or 9,000 tags per minute (the minimum location rate). The ranging procedure starts with the tag listening to anchor messages, in order to discover the network topology and chose the reference anchors. By default, the multilateration algorithm is carried out by the tag once the anchors have been assigned their relative positions in a xyz-grid. Maximum likelihood is applied in the multilateration algorithm and the estimated tag position is obtained after a moving average

(over five estimations) filtering. The returned position refers to the anchor supposed to be at the (0,0,0) position and has millimeter resolution.

2.3.3 The joint GPS-RTLS enabled LoRaWAN node

A proof of concept demonstrator has been realized for evaluating the proposed joint GPS-UWB-RTLS localization solution for LoRaWAN networks. Both the tag and the anchors are built around an (oversized) Raspberry Pi 2B, hosting a Broadcom BCM2836 (which includes a quad-core Cortex-A7), as shown in **Figure 2-20**. The Raspberry runs the LoRaWAN protocol stack, which is a porting of the IBM LMiC stack, compliant with Class A LoRaWAN specifications. The LoRa physical layer is implemented by a SX1276 radio from Semtech hosted in a RFM95 module from HopeRF and connected via an SPI link.

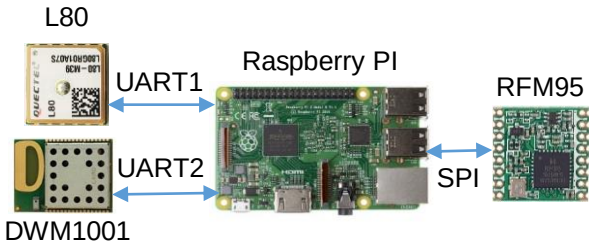


***Figure 2-20**: Block diagram of the proposed indoor/outdoor tag*

Localization is performed by intelligent peripherals connected to the Raspberry using USB-to-UART cable converters. In particular, the outdoor localization exploits the MTK MT3339 device from Mediatek, hosted in a L80 module from Quectel. It provides a superset of the National Marine Electronics Association (NMEA) protocol via UART, that can be used for retrieving localization and timing information. Regarding the indoor localization, the Raspberry is connected with the aforementioned DWM1001 module from DecaWave; in particular, the nRF52832 microcontroller executes the DecaWave location firmware and manages the communication protocol over the UART link with the Raspberry host. Tag power supply is furnished via a power bank; consumption has not been optimized and both GPS and UWB devices are always active. The Raspberry acquires both outdoor and indoor localization data, which are encapsulated into a LoRaWAN message sent every minute.

2.3.4 Experimental Results

The aim of this section is to describe the results obtained thanks to the realized experimental test bench. Experiments were carried out in the engineering campus at the University of Brescia. The LoRaWAN coverage of the campus area is provided by a public infrastructure,

managed by the A2A Smartcity company, a member of the A2A multi-utility operating in the area of Brescia. The LoRaWAN backed is the PTNetSuite. LoRaWAN messages contain localization data as provided by the intelligent peripherals, i.e., in terms of absolute latitude and longitude for outdoor localization (two 32-bits signed numbers in format ±(d)ddmm.mmmm, degree and minutes) and as a relative position with respect to the anchor grid (one 32-bits signed number per each coordinate) for indoor localization. Consequently, the LoRaWAN message payload is 32 B and consists of the transmission timestamp (as UNIX epoch time in milliseconds, 8 B); of the latitude and longitude from GPS (8 B) and of the RTLS coordinates (12 B) as provided by the location engine embedded into the tag. When the 36 bits of the physical layer and the 13 B of data link headers and trailers are considered, the resulting overall time on air is about 92 ms since SF = 7 and CR = 4/5 is used (fastest data rate).

In the measurement campaign, a similar setup has been adopted for both indoor and outdoor scenarios. Two (outdoor) or four (indoor inside a shed) reference nodes (anchors) have been located on vertexes of a square and the data coming from a fixed tag have been collected (see **Figure 2-21**). In all cases, the tag and the anchors lay 1m above the floor and a planar (2D) configuration is considered. Anchor A0 corresponds to the origin of tracking area reference system; anchor A1 defines the x-axis, whereas (if present) A2 is positioned along the y-axis.

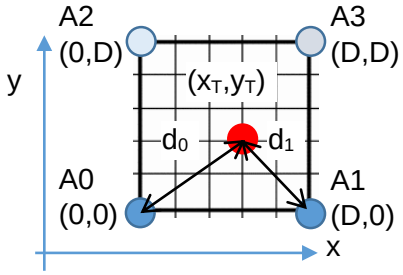


Figure 2-21: *The proposed experimental setup. Four anchors (dark and light blue) are used indoor (D = 10m), two (dark blue) outdoor (D = 60 m).*

For sake of simplicity, the actual tag coordinates (x_T, y_T) have been estimated using the minimum required set of anchors (i.e., A0 and A1, since ambiguity is a-posteriori resolved discarding the wrong solution) using trilateration as in (2.1):

$$\begin{cases} x_T = \frac{D}{2} + \frac{d_0^2 - d_1^2}{2D} \\ y_T = \sqrt{d_1^2 - {x_T}^2} \end{cases} \qquad (2.1)$$

The set of distances $\{D,d_0,d_1\}$ is obtained via measurement performed by a Fluke 414D laser distance meter, which has an uncertainty better than 0.15 mm/m in its useful range (< 100 m). As a result, the uncertainty in the actual tag position is negligible with respect to the location error *E*, evaluated as in (2.2):

$$E = \sqrt{(x_M - x_T)^2 + (y_M - y_T)^2} \qquad (2.2)$$

where x_M and y_M are the values returned by UWB or GPS techniques. In particular, the pair (x_M, y_M) for the GPS scenario is obtained post processing the latitude and longitude coordinates (natively furnished by the L80 module) according to a Flat Earth model, in order to refer to a planar coordinate system.

The two modules show very different behaviors, as confirmed by an excerpt of 400 samples acquired every second, when a tag is located in fixed locations, for both indoor and outdoor scenarios in **Figure 2-22**. In particular, it can be noted the L80 module applies a dead band algorithm, so that deviation in the position value occurs only after exceeding a threshold value. For sake of simplicity, if the position value is not changing, measurements are supposed uniformly distributed within the dead band (estimated to be about 5 m), and their standard deviation has been evaluated accordingly. The actual tests have been carried out in optimal (no obstacles between anchors and tag indoor, clear view of the sky outdoor) and sub-optimal (obstacles in the area delimited by anchors indoor and partial view of the sky outdoor) conditions for both scenarios and lasted 3 hours, i.e., 180 messages have been considered, since the LoRaWAN update rate was one message/min. Regarding GPS, the number of tracked satellites ranges from 6 to 9 and 15 minutes before starting the experiment have been waited for ensuring fixing. Obtained results are reported in Table 2-12; they are in agreement with other similar research available in literature [2.66],[2.67] confirming the feasibility of the proposed approach.

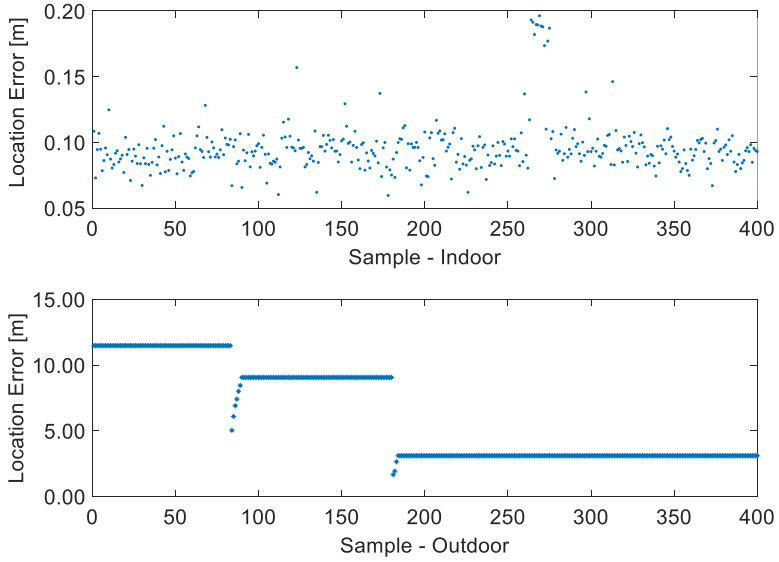


***Figure 2-22**: Example of distance error for indoor (Site 1) and outdoor (Site 1) scenarios.*

Table 2-12: *Location errors for indoor and outdoor scenarios.*

Measurement	D_{AVG} (m)	D_{MEDIAN} (m)	D_{STD} (m)
UWB-RTLS Indoor – Site 1 **No obstacle in the area**	0.09	0.09	0.02
UWB-RTLS Indoor – Site 2 **Obstacle in the area**	0.38	0.36	0.04
GPS Outdoor – Site 1 **Clear view of the sky**	6.2	3.1	3.6
GPS Outdoor – Site 2 **Partial view of the sky**	20.7	17.2	13.5

2.3.5 Final Considerations for LoRaWAN applications in a Smart City context

In this section, the jointly use of LPWAN and RTLS solutions is suggested for Smart Campus applications. In particular, the LoRaWAN has been selected for implementing the communication infrastructure, thus enabling indoor and outdoor coverage using a single technology. Regarding the localization, a low-cost GPS module has been used outdoor, whereas a UWB-based device has been used indoor. Experimental results confirmed the feasibility of the proposed approach. Measurement campaigns highlighted location error in the order of few tens of meters outdoor (e.g., due to proximity to buildings), whereas indoor location errors on the order of few tens of centimeters are obtainable. New experiments will be carried out for the GPS scenario exploiting improved anchors location estimation e.g. using an approach similar to [2.68]. Additionally, if indoor and outdoor localization data are simultaneously available, accuracy could be improved performing data fusion algorithms in a cloud-supported cyber–physical localization framework for Smart Campuses [2.69].

2.4 LoRaWAN infrastructure for Industry 4.0

This research deals with the creation of the prototype of an embedded platform which includes both OBD-II interface, 3G/4G connectivity and also LoRaWAN for backup situation. Considering the Smart City use cases, the application constrains regarding intelligent vehicles are discussed in order highlight design directions for the correct integration with the LoRaWAN infrastructure.

2.4.1 Technology Overview and Objectives

The Internet of Things, Industry 4.0 and Intelligent vehicles paradigms are pushing the introduction of several new technologies. In the following, a short overview of the technologies

relevant for this section is given, followed by a state-of-the-art analysis and the list of the section goals.

2.4.1.1 OBD-II

The OBD-II is a device attached to the vehicle's ECU (Engine Control Unit). It provides a communication interface for exchanging the data provided in real-time by vehicle sensors and actuators. The communication between OBD-II and the car ECU is possible through many different protocols. Normally, each vehicle carries just one protocol and the most used one is the Controller Area Network (CAN), developed by Bosh in 1980. However, there are also others protocols such as SAE J1850 PWM, SAE J1850 VPW, ISO 9141-2.

The OBD-II interface has ten operating modes in order to fulfill specific purposes. There are distinct sets of commands used to extract data from vehicle sensors and actuators 2.69, by using unique codes known as Parameter IDs (PID). It is not compulsory that a manufacturer implements all functionalities in all its vehicles. Thus, it is important to check which commands the vehicle supports before interactions.

The data acquisition and analysis system considered in this section is attached to the vehicle OBD-II port and set up to send the stored data to the cloud database through the “currently” available network connection. Related works and objectives

The scientific literature regarding the Internet of Intelligent Vehicles, the use of LoRaWAN for vehicles and the inclusion of vehicles into the Smart City ecosystem, is increasing.

A fleet management system based on OBD-II is carried out in 2.70. An OBD-II reader is used to collect the speed and mass air flow data, from which the distance and fuel consumption are also measured.

A customer feedback platform for the automotive industry in Industry 4.0 area is illustrated in [2.70]. Such platform is able to get and examine data from vehicle sensors in order to manage the control, prevention, and mitigation of car problems. It transmits messages collected by an OBD-II scanner.

Moreover, [2.72] suggested a graph database solution (oriented to Industry 4.0) to manage the enormous quantity of data created by the sensors spread inside the vehicle in an IoT scenario. Some benefits were acknowledged through the vehicles’ geo-localization and discovered events.

Furthermore, other solution [2.73], aims to evaluate the quantity of carbon dioxide emission (based on vehicle sensors) to control the vehicular pollution. Despite that, it was not applied to the Industry 4.0 scenario.

Electrical vehicles will be a key part of future mobility scenarios inside Smart Cities. In [2.20], a detailed analysis of communication requirements for the management of recharging station is reported; 4G and LoRaWAN are considered and compared.

The coverage of 3G/4G inside urban area may be unpredictable. In [2.74] some of the authors analyzed the current situation in a large city in Brazil, finding evidences of lacks in the coverage when 3G devices are onboard moving vehicles (in real traffic conditions).

Last, the coverage of wide area by means of LPWAN infrastructure is a reality in several cities. For instance, the city of Brescia (Italy) has one of the most complete LoRaWAN infrastructure in Europe, as described in these works [2.75],[2.52].

Based on the previous literature, the main goal of this section is to provide a more flexible, OBD-II-based solution for vehicle monitoring oriented to Industry 4.0 applications. The new architecture should take into account, and opportunistically use, all the available communication channels available in modern Smart Cities, in order to assure the flow of information. In particular, LoRaWAN technology has been evaluated to be the most viable solution.

2.4.2 The Proposed Prototype

The proposed prototype exploits a suitable backup network to overcome possible lack of coverage of the 3G/4G infrastructure. The design goals are

- Long range
- Capability of transmission of mandatory data to assure continuous monitoring of vehicles/driver activities
- Real-time transmission with reduced latency

The proposed architecture is based on a platform previously evaluated at the University of Natal [2.74], combined with the long range capability of LoRaWAN protocol.

In detail, the architecture is shown in **Figure 2-23**. The data coming from the OBD II interface are read using an AutoPi edge device with OBD II capability. The LoRaWAN transmission is obtained by means of a LoRaWAN modem.

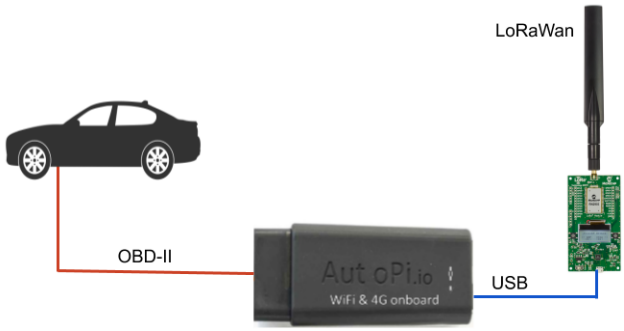


Figure 2-23: Proposed module integration for LoRaWAN transmission as backup to standard WiFi and 3G/4G.

2.4.2.1 The design of the prototype

The AutoPi device is a true Edge computing device, since it is equipped with a Raspberry Pi Zero module (1GHz single-core CPU, 512MB RAM). The device is running a Raspbian Linux version. The LoRaWAN interface has been realized with a python software with *miniterm* serial tool connected, through an USB port, to the Microchip LoRaWAN Module RN2903 from Microchip (operating in the 915 MHz ISM band) with LoRaWAN 1.0.1 stack. (Note: Microchip RN2483 Module can be also used, if the prototype is operating in the European band at 868 MHz)

The most critical part of the prototype design was the definition of the data to be transmitted by means of LoRaWAN, since there are Regional packet size limitations and duty cycle limitations, as previously stated.

The data inside the vehicle are grouped by PID, as discussed before. The size of each PID is greatly variable and very often includes the timestamp. The most important PIDs are: latitude, longitude, speed, rpm, throttle position, accelerometer, Mass Air Flow (MAF) and Manifold Air Pressure (MAP); with these PIDs is possible to monitor vehicular emissions and driver behavior. Table 2-13 lists the size in byte (B) of these PIDs. The payload size of LoRaWAN messages is varying with the SF coding, but in order to pack as much information as possible, only the largest LoRaWAN packets can be used. In particular, the payload transmitted is 159B long (including only the timestamp of the Latitude), allowing only SF7 and SF8 to be used.

The maximum payload and the duty cycle limitations vary from country to country. The more restrictive duty cycle of 1% is considered in this section in order to maximize compatibility; for the same reason, SF7 and SF8 @ BW=125KHz have been addressed. Since the LoRaWAN packet with 160 byte payload has a time on air of 262ms@ SF7 and

462ms@SF8, the maximum update rate with a backup LoRaWAN connection is 26 s and 46 s (satisfying the requirements of 20 s interval between successive uplinks of Brazilian regulations). This situation is very far from the usual 3G/4G case, where update rate that is in the order of few seconds.

Table 2-13: *Details of PIDs transmitted inside the LoRaWAN frame. (all sizes are in bytes.)*

PID	Size	Size with timestamp
Latitude	12	44
Longitude	10	42
Latitude + Longitude	20	52
Speed	5	37
Engine RPM	6	38
Throttle	20	52
MAF	10	42
MAP	10	42
Accelerometer	34	66

2.4.3 LoRaWAN infrastructure in Smart City

In this section, an example of real LoRaWAN infrastructure installed in a Smart City is presented. The city of Brescia (Italy) installed in the last three years one of the wider urban network based on the LoRaWAN technology. The LoRaWAN infrastructure is used for handling primary services like: district heating, waste collection and handling, water/energy consumption monitoring (especially in the university Smart Campus [2.52]). Up to the current date, 123 LoRaWAN Gateways have been installed in a densely urbanized area of more than 75 km^2, resulting in an average density of 1.6 Gateways/km^2. The geographical distribution of the deployed gateways is shown in **Figure 2-24**, where it is visible that the density is not uniform, because the density of gateway is also proportional to the density of end-nodes in the surroundings. Hence, most populated areas contain more end-nodes, in turn requiring more gateways. In order to evaluate the minimum radius of the cell covered by one gateway, the minimum distance between a gateway and its neighbor gateway has been calculated. **Figure 2-25** shows the distribution of the minimum distances, highlighting that most of the gateways are placed about 400 m apart from the closest one. Average distance is 500 m, minimum distance is 110 m and maximum distance is 1500 m.

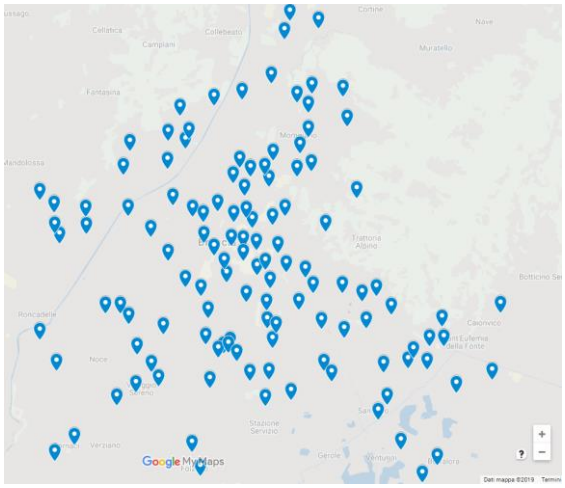


***Figure 2-24**: Geographical distribution of LoRaWAN Gateway in the city of Brescia.*

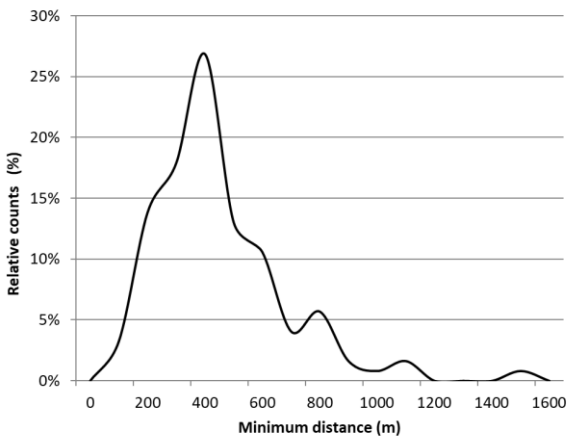


***Figure 2-25**: Distribution of the minimum distance between any two LoRaWAN Gateway in the city of Brescia*

2.4.4 Smart City Use Cases

In this section, the constrains derived from use cases in Smart City scenario are introduced. In particular, the experiments are oriented to: evaluate the probability that an end-node is able to send its data to the considered infrastructure; and the distance between the node and the base station to have successful transmission. Then, some design directions for proper integration of intelligent vehicles with opportunistic LoRaWAN capabilities are given.

2.4.4.1 Characterizing the 4G coverage

The characterization of the 4G coverage was done with experiments performed on an urban route in the city of Natal, Brazil. The route is approximately 5km long and was run in two different days between 9AM and 10AM. For each day of the experiment, four rounds have been completed with the cars listed in Table 2-14. The experiment was carried out on these

schedules in order to obtain more variability in the data. Each vehicle were equipped with the prototype described in the previous section. In the AutoPi device the parameters were set as follows: connection type to 4G, with APN Tim; data receiving server hosting on AutoPi Web Service; PID set request interval configured to 1 s. In this way, data was collected for about 20km, with 5km on each round of the experiment per vehicle. The data were collected on urban roads, with total duration of 40 minutes and an average of about 10 minutes per cars. The graphical view of the route is shown in **Figure 2-26**, where the dots represent the successful transmissions carried out by the AutoPi 4G interface.

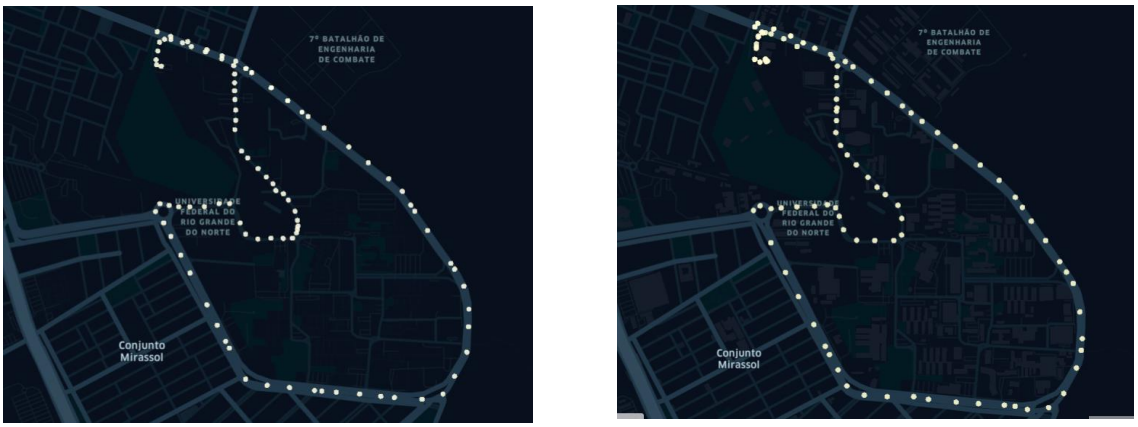
***Table 2-14**: Cars used in the considered urban scenarios in the city of Natal (Brazil)*

Model	Motor	Transmission	Year
Ford KA	1.5	Automatic	2019
Chevrolet ONIX	1.4	Automatic	2015

The results of the experiments are shown in Table 2-15.The failure rate (number of failed attempts over the total number of connection attempts) is in the order of 10% when the car is moving.

***Table 2-15**: Failure rate of 4G connection in the considered urban scenarios in the city of Natal (Brazil)*

Model	Sample	Sample Fail	Sample Success	% Fail	% Succ.
KA – day 1	109	10	99	9	91
KA – day 2	120	11	109	9	91
ONIX – day 1	106	11	95	11	89
ONIX – day 2	120	16	104	13	87



A) **B)**

***Figure 2-26**: Car itinerary for the considered urban scenario in the city of Natal (Brazil). a) Ford KA, b) Chevrolet ONIX.*

2.4.4.2 Characterization of LoRaWAN coverage in Smart City

The coverage of LoRaWAN in case of fixed end-nodes has been experimentally evaluated in Natal (Brazil). The case of mobile nodes has been tested in Brescia (Italy).

2.4.4.2.1 Fixed end-node

The experimental setup is composed of a LoRaWAN gateway, placed at the UFRN Industrial Informatics Laboratory, and a LoRaWAN end-node, acting as a sensor. The LoRaWAN gateway is a Sentrius RG191 device with the following characteristics: two antennas dual band WiFi, one LoRa antenna, and one Ethernet port (RJ45). The gateway can be interfaced with many LoRaWAN backend implementations: Semtech, LoRaIO, The Things Network (TTN), Stream, etc. The LoRaWAN end-node is realized using a Microchip RN2903 module which has the following characteristics: Class A LoRaWAN implementation, support to the 915 MHz ISM band, and ASCII modem interface.

The LoRaWAN end-device is placed at 20 points inside the campus and it sends packets containing a 50B long payload. The worst-case condition from the sensitivity point of view has been chosen: messages have been sent using SF7 @ C/R = 4/5. The transmission power has been set to +14 dBm. The tested locations have been chosen by increasing the distance between the gateway and the LoRaWAN mote. **Figure 2-27** shows the mote locations, identified by blue marks (if the transmission was successful) or red marks (when transmission was not successful), as well as the Gateway position, identified by the purple cross. At each chosen location, the test procedure has been repeated fifty times.

The real operating distance resulted to be up to 250 m from the gateway, when this urban scenario is considered (and LoRaWAN operates with the weakest coding SF7). The presence of tall buildings and complex terrain explains the partially reduced coverage area.

2.4.4.2.2 Mobile end-node

Also mobility scenarios were already evaluated for LoRaWAN inside the Brescia Smart Living project [2.52]. The LoRaWAN gateway is the Sentrius RG186 device while the LoRaWAN end-node is built with a Microchip RN2483 module. A car equipped with the LoRaWAN transmitter was travelling along an itinerary of about 4 km at the maximum speed of 50 km/h and an average speed of 17 km/h. All the packets sent were captured by gateways belonging to the Smart City infrastructure described in the previous section, confirming the usability of LoRaWAN for urban mobile applications.

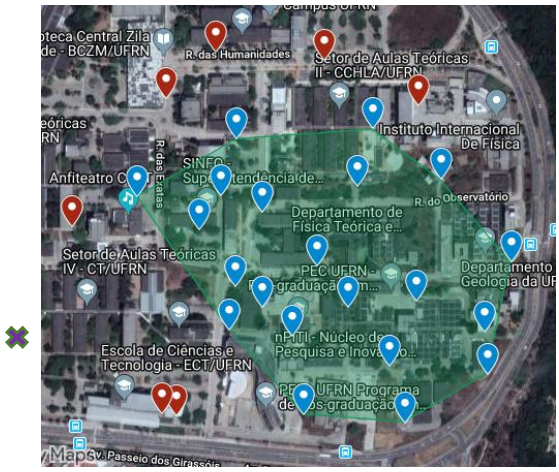


Figure 2-27: *Geographical distribution of the locations covered by a LoRaWAN Gateway in the UFRN campus in the city of Natal.*

2.4.4.2.3 Review and design directions

The results previously discussed confirm the feasibility of the proposed approach. In particular, the experiments allowed to identify some key points.

It must be underlined that the need for transferring relatively large amount of user data (at least if compared with the message length of LoRaWAN intended applications, as smart metering) complemented by fast update rate for ensuring real-time behavior, forces the adoption of the fastest, (but less noise/interference immune) data rates supported by LoRaWAN. In turn, this implies that, when an urban scenario is considered, the (mobile) node must have a distance from the nearest gateway on the order of 250-300 m when it is working with SF7. These types of deployments can be considered relatively dense, but they have been already carried out in many places. The suggested deployment density is about 1.5 Gateway/km^2.

Additionally, the limitations on transmission of successive messages strongly affect the refresh rate. Considering the use of SF7 and SF8 @ BW=125kHz and d=1% (permitted in both the Europe and Brazil regions and allowing the desired 160B-long payload), the suggested refresh rate is in the order of 30 s and 60 s, respectively.

It has to be noted that in other countries a maximum dwell time (or a different minimum interval between successive uplinks) could be specified. Nevertheless, a slower update than 4G scenario must be taken into account and handled at application level.

2.4.5 Final Considerations for LoRaWAN infrastructure for Industry 4.0

Intelligent vehicles are the future of the automotive sector, which is built over Industry 4.0 paradigms. Car manufacturers are already collecting data from vehicles connected to Internet

for improving production (and products) during the entire lifetime of plants (and products). Today, the OBD-II interface is the access point (to onboard sensors) in many cars and trucks. In order to increase the number of connected vehicles, devices with OBD-II and 3G/4G modems are used, even if there are situation where the 3G/4G coverage/capability in not sufficient. The original idea of this section is to exploit Smart City infrastructure as a backup connection for intelligent vehicles that temporary lose their main 3G/4G link. Smart Cities are heavily based on data sharing, and specific communication infrastructure have been deployed. In particular, new wireless technologies are widely used, as for instance LoRaWAN. This section presented a prototype of an embedded platform which includes both OBD-II interface, 3G/4G connectivity and also opportunistic LoRaWAN connection for backup situation. It is considered Smart City use cases regarding intelligent vehicles, deriving the application constrains that can be used for the correct LoRaWAN integration.

2.5 Final Considerations for LoRaWAN Architecture

These researches mentioned in this chapter present a deep investigation and validation of LoRaWAN Architecture, first a methodology for the experimental assessment of architectural delays of LoRaWAN implementations in private and public networks. In the latter, the experimental results demonstrates the proposed methodology is efficient to compare the time-delays performance of Semtech and LoRa Server implementations into a local network. The specific details of the measurements can be used to make better decisions or to increase results for further optimization actions. Related to public LoRaWAN infrastructure, using the Patavina Netsuit from A2A Services, a experimental setup, synchronized by means of the NTP protocol, has addressed to validate the complete remote end-to-end delay chain from the source and the final user of data. The average overall delays vary between 400 and 700 ms, but abnormal peaks of several seconds have been recorded as well.

Moreover, continuing to stress LoRaWAN technology, the jointly use of a LoRaWAN modem and localization engines, a low-cost GPS module (outdoor) and UWB-based device (indoor), is suggested for Smart Campus applications. The measurement campaigns confirmed the feasibility of the proposed approach. The results demonstrated location error in the order of few tens of meters outdoor (e.g., due to proximity to buildings), whereas indoor location errors on the order of few tens of centimeters are obtainable.

Finally, the LoRaWAN network is used as backup network in case of lacking of 3G/4G connectivity for Industry 4.0 scenario using Intelligent Vehicles. This section studied an

AutoPi device, with OBD-II Interface and 3G/4G connection, embedded with a LoRaWAN modem. This research demonstrated the application constrains and design directions to achieve a correct integration between LoRaWAN infrastructure and the Internet of Intelligent Vehicles for Industry 4.0.

The deep investigation and validation of LoRaWAN network addressed the feasibility of this promise technology to be used as an communication infrastructure over large area in IoT scenario, but some limitations and constraints are still demanding some researches and answers. In the next chapter, some studies were conducted to enhance LoRaWAN network to mitigate some weak points inside its architecture.

3 Enhancing LoRaWAN Architecture

After the deep research in LoRaWAN architecture in chapter 2, new researches are addressed and evaluated to enhance LoRaWAN architecture. First, an innovative LoRaWAN mechanism is proposed and tested for extending LoRaWAN coverage in an Industrial IoT scenario. Moreover, a novel LoRa-REP access method validates a redundant transmissions strategy to improve success probability in emergency situations.

3.1 Objectives

The studies conducted in chapter 2 demonstrated the applicability to eHealth of the promising LoRaWAN technology. Besides the good results extracted from experimental campaigns, there are some constraints in LoRaWAN technology: network coverage and transmission delivery rate. LoRaWAN range is in order of few kilometers, but poses some restrictions in indoor coverage (e.g. inside hospitals, facilities and factories) and also the delivery ratio of data packet is still not enough, frequently below 70%, which not satisfies applications in emergency situations. For these reasons, this chapter aims to enhance LoRaWAN architecture by means of:

- **LoRaWAN Range Extender**: developing and evaluating an extender node based on commercially available device, named as e-Node, to increase LoRaWAN coverage in a real use case scenario. This proof-of-concept prototype, which is transparent to network, is in charge of replicating LoRaWAN messages.
- **Improving LoRaWAN feasibility in Emergency situations**: using the same e-Node from LoRa Range Extender, a LoRA-REP access method was proposed to enhance LoRaWAN Technology in emergency situations. In order to evaluate its feasibility, the tests were performed using fixed and mobile positions in public and private LoRaWAN networks.

In order to clarify the Chapter 3 workflow, the **Figure 3-1** demonstrates its flowchart.

3.2 LoRaWAN range extender

Despite LoRaWAN promises long range and dense environments with many obstacles (as the industrial ones), it may suffer from coverage issue. Additionally, the inverse relationship between data rates and range may be unacceptable for many industrial applications. In this section, an innovative LoRaWAN range extender (based on an enhanced LoRaWAN node) is described, and its integration in the infrastructure of industrial Internet of Things (IIoT)-enabled industrial wireless networks is presented. A frame relay strategy is suggested, thus avoiding to tradeoff the highest data rates against an increased sensitivity. The feasibility of the solution is formally verified and the features of the realized proof-of-concept prototype, based on commercially available hardware, are discussed. The experimental results, obtained with a purposely designed test bench, show the effectiveness of the proposed approach. In particular, the range extender is transparent for legacy LoRaWAN networks as confirmed by correct relaying of uplink and downlink for different types of LoRaWAN message.

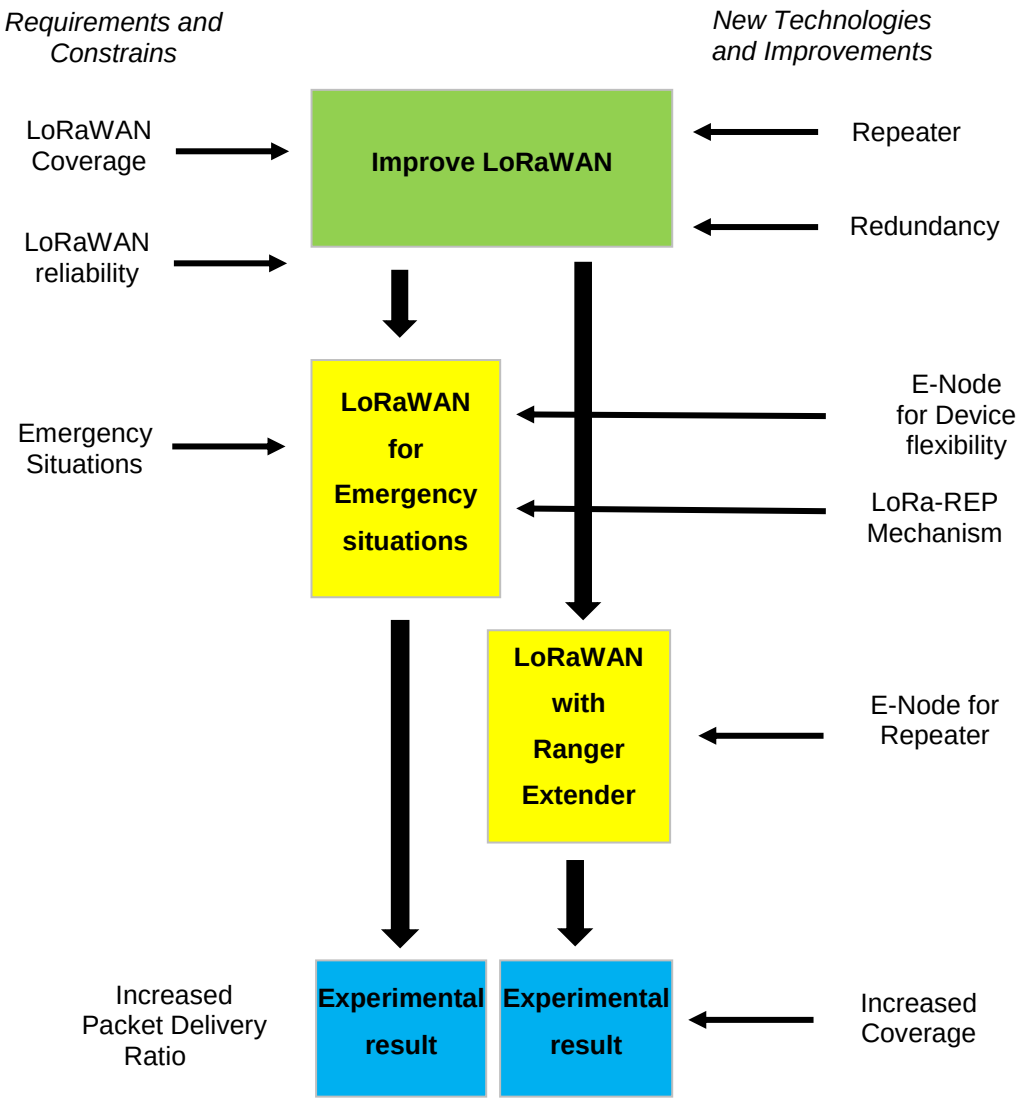


***Figure 3-1**: Graphical abstract of Chapter 3.*

The idea of a (transparent) relay node, called e-Node, suitable for LoRaWAN has been introduced in [3.1]. In this paper, the e-Node embryonic idea has been validated and changed into a proof-of-concept prototype, tested in a vast measurement campaign. The e-Node has been developed having industrial applications in mind, since it operates with the highest LoRaWAN data rates, thus preserving the communication timeliness. In particular, the main original research contributions are:

- the theoretical validation of the proposed approach;
- the implementation of a proof-of-concept prototype implementing the full set of capabilities of the proposed relay node;

- an extensive measurement campaign based on a purposely designed test bench mimicking real-world networks.

The transparent interoperability of the e-Node with an existing network is demonstrated, since both regular up/downlink and join procedures are supported. The proof-of-concept prototypes are based on commercially available hardware. Each one includes: two wireless frontends composed of RaspberryPi 2 connected to Laird RG186-M2 module; and one processing unit (both a PC and a Linux-based embedded system IOT2040 from Siemens have been tested).

3.3 Motivations and Comparison with existing solutions

The IoT paradigm poses requirements and faces some constraints that legacy communication networks are not able to satisfy. Consequently, new technologies emerged, especially for wireless communications. These solutions ensure autonomous operation and long lifetime since, in most of cases, relatively long update time is tolerated. While waiting for the wide adoption of 5G, LPWAN technologies appeared as other solutions for creating private cellular-like networks with minimal infrastructure costs. Among others, the LoRaWAN solution, an open standard managed by the LoRa alliance, gets attention thanks to the availability of modules from many manufacturers. The long-range capability of LPWANs suggests that a reduced number of wired gateways must be deployed on the field for covering wide areas, thus lowering both initial and maintenance costs. Such an advantage is attractive in various application domains, as investigated by many researchers [3.2][3.3], including the industrial one [3.4][3.5]. However, despite many scenarios are homogeneous (e.g. purely indoor applications [3.6]), the cellular-like architecture may result in heterogeneous indoor/outdoor applications. In such scenarios, evaluating the available link budget can be difficult and the actual transmission range can be smaller than expected due to the possibly higher attenuation factor, especially when industrial applications are considered [3.7]. Generally speaking, it is possible to define a breakpoint distance separating a (closer to the transmitter) region where the propagation is in line-of-sight (LOS) from the rest of the area exhibiting non-line-of-sight (NLOS) behavior. In the former case, the path loss exponent is two, while in the latter can be four or even higher [3.8].

Referring to LoRaWAN, specifications define restrictions on the maximum payload size for possible encapsulation overhead, but repeaters are not defined. Some solutions available in literature propose the use of tunneling across different wireless wide area networks, thus extending the coverage at the price of increased complexity and cost (e.g. leveraging on mobile

communications as in [3.9]). Some works describe possible enhancements for supporting multi-hop topology [3.10]-[3.12]. However, these works do not ensure backward compatibility with existing LoRaWAN infrastructure, therefore they can be considered like brand new protocols above the LoRa radio. In other works, the compatibility is not transparent. For instance in [3.13] an "hybrid node" uses the ABP provisioning for interacting with nodes on the field on one side and adopts OTAA for connecting to the LoRaWAN infrastructure on the other side. In this case, two major drawbacks can be highlighted: the local ABP activation poses additional maintenance issue; and the high complexity of the node. There are some commercial products that actually implement repeaters using proprietary solutions above LoRa radios. A couple of examples are the "LoRa RF Repeater" from Invisible systems and the "INS-extender" from Ineo-Sense.

Relay (or helper) nodes, exploiting the broadcast nature of wireless communications to extend range and coverage without affecting the (wired) infrastructure, could be a viable solution for solving this challenge. However, according to the best of the authors' knowledge, the design, implementation and characterization of a pure frame relayer operating at Layer 1/ Layer 2 and fully compliant with the LoRaWAN specifications has not yet been described in literature. This section tries to fill this gap, confirming feasibility and advantages offered by such a relayer.

3.4 The proposed LoRaWAN range extender

As previously stated, it must be noted that the virtual channels obtainable by varying the SF value have different processing gains resulting in different reception sensitivity and thus broader ranges. Unfortunately, over-the-air duration of messages become longer and longer, due to the exponential growth in symbol length, increasing both the probability of a collision with an interferer and the side effects of clock drifts. Thus, overall performance may be greatly affected by SF allocation. The simplest approach for resource assignment would allocate the lower SFs to nodes having better RSSI (received signal strength indicator) at the gateway (because they are reasonably closer), and higher SFs to nodes far away from the gateway. Unfortunately, this simple heuristic method suffers from the aforementioned exponential growth of message length transmitted with higher SFs, which can be limiting in industrial applications having timing constraints. For this reason, more sophisticated heuristics have been proposed, as the EXPLoRa-AT; indeed in [3.14] it is suggested a fair allocation mechanism trying to equalize the network load using faster SFs for longer messages and vice versa. The

price to pay is a non-uniform coverage, since only nodes close to the gateway can send long messages.

Resuming, hybrid and heterogeneous indoor/outdoor scenarios pose a coverage issue [3.7], which is not strictly related to the distance from the gateway, but depends on the (usually *a priori* unknown) characteristics of the radio link. The trivial solution of denser gateway displacement is often not feasible, since it results in an increase of the (typically wired) infrastructure complexity, with unjustifiable cost for most of intended applications. A more affordable and effective solution could be the adoption of a so called frame relay or helper node, that leverages the broadcast nature of the wireless medium and performs a retransmission on behalf of the source. Such a device is nothing but a modified wireless node and does not require a connection with the backbone, so that it can be easily deployed where signal enforcement is required (e.g. after a survey) [3.15]-[3.18].

In this section, authors propose the adoption of an enhanced LoRaWAN node (e-Node in the following) that can extend the coverage of messages transmitted using the highest data rates. Only main power supply is required, in order to ensure the asynchronous reception of incoming messages. The e-Node is different from the "repeater" described in the LoRaWAN specifications, which is defined as a device "using LoRaWAN as its backhaul mechanism".

3.4.1 Retrieving information from over-the-air LoRaWAN frames

As previously stated, the application payload is encrypted using the so called AppSKey obtained after the activation procedure. Additionally, we assume the e-Node is a standalone node-in-the-middle; it is not connected to the backend infrastructure and thus it is not able to retrieve keys from the JS. This approach ensures that the overall network security is not compromised when the node is removed (e.g. stolen) from the field. Since the e-Node operates at the data link level, it is useful to understand what frame fields shown in **Figure 3-2** are not enciphered and available for further processing, most notably the MAC Header (MHDR), the Frame Header (FHDR) and the Port Field (FPort):

- the MHDR specifies the message type (MType) and the major version of the LoRaWAN specifications used for frame encoding;
- the FHDR contains the short device address of the end-device (DevAddr), a frame control octet (FCtrl), a 2-octets frame counter (FCnt), and up to 15 octets of frame options (FOpts) used to transport MAC commands. The FCtrl contains the acknowledgment bit (ACK) which is set by the sender of a confirmed message. The FCnt field is actually the least significant word of the counter tracking uplink and

downlink messages, after the activation procedure. Since it is used for enciphering the user payload, permits to detect reply attacks;

- the FPort allows to distinguish between LoRaWAN command endpoints versus implementation specific application endpoints.

When behaving as a single-hop range extender, the proposed e-Node only relies on MType for distinguishing between *join_request*, *join_accept*, *confirmed* and *unconfirmed* messages. If the downlink is required, the DevAddr (obtained in the join procedure), is used for pairing with the homologous uplink. Interesting to highlight, the DevAddr is used for ciphering the payload, thus the implementation of a routing protocol above LoRaWAN for OTAA networks is not simple.

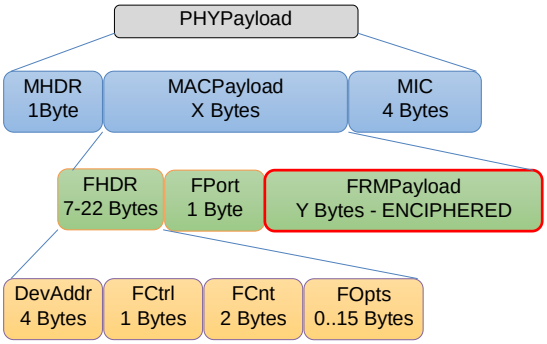


Figure 3-2: *LoRaWAN message fields (PHY header and trailer, not shown, depends on LoRa; the FRM payload is encrypted using the AppSKey session key univocally assigned to each end-device).*

3.4.2 Functionalities and limitations

The e-Node has been designed in order to act as a transparent range extender; it is of primary importance that it coexists with and it does not affect functionalities of regular node on the field. Accordingly, its operation is limited to the DL layer preserving full compatibility with the LoRaWAN specifications. For this reason, only single-hop extension has been considered [3.18 , 3.19].

Implementing a routing layer above the LoRaWAN stack (e.g. for multi-hop forwarding) would require modification in the backend servers. On the contrary, the proposed e-Node only listens at uplink and downlink messages, which are propagated to the final destination. In particular, the e-Node takes advantage of the NS capability to deduplicate the same message received from multiple co-located gateways, accepting only one (not necessary the first arrived). In this way, no additional mechanisms are required by the proposed e-Node, even in case both the original and the repeated messages are received by the gateway(s). The difference

respect to regular nodes is the need of wired power supply for continuously listening at the medium.

The main limitation of the e-Node comes from the regional regulation bodies that often impose duty cycle limitation for the frequency bands used by LoRaWAN. For instance, in EU imposed duty cycle (i.e. the ratio between the transmitting time and the time between transmissions) varies between $d_{min} = 0.1\%$ and $d_{max} = 10\%$ depending on the frequency channel. The transmitting time of the e-Node must take into account both uplink and downlink forwarding, no matter if the latter is only an acknowledgment for a confirmed message. Fortunately, the most messages sent on LoRaWAN networks are unconfirmed uplinks that are not tied to the time constrains of downlink. Hence, suitable delayed transmission strategies can be used in the e-Node to mitigate the impact of duty cycle limitations.

From the scalability point of view, adding an e-Node (like adding any other node) worsens the network goodput, due to possible collisions deriving from the simple ALOHA medium access strategy of LoRaWAN, a well-known result in literature [3.21]. However, the e-Node is intended only for the highest data rates (i.e. the messages are shorter), minimizing the possibility of overlapping in time and frequency.

3.5 The range extender core algorithm

Every time a new uplink is captured, the e-Node replicates it; for unconfirmed message the sending time is not relevant and no limitations exist. On the contrary, if a subsequent downlink is required, the two receiving windows (see **Figure 3-3**) must be used and synchronization between the up and down paths is required.

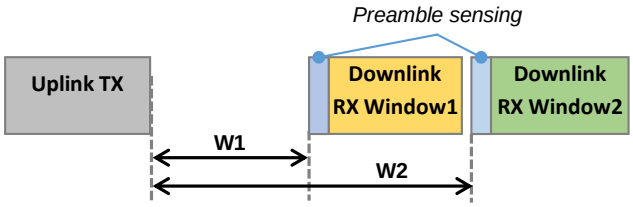


Figure 3-3: *LoRaWAN Class A Transaction. The transaction is node-initiated (uplink direction): the node wakes up and transmits its message; two receive windows after W1 and W2 allow for downlink.*

In **Figure 3-4**, the regular behavior of a LoRaWAN transaction is shown. The e-Node (grayed out) repeats the uplink after the F delay is expired. The message will be successively discarded by the deduplication mechanism of the NS. The uplink over-the-air message duration is M_{NU}, while the downlink over-the-air message duration is M_{GD}. Typically, W1 = 1 s for data messages and W1 = 5 s for *join_request*, W2 = W1 + 1 s.

If the node is far away from the gateway, only the repeated uplink messages is received by the NS, as shown in **Figure 3-5**. If a downlink transmission exists, it should be sent by the gateway after W1 from the end of the repeated downlink. A NS supporting only the second receiving window is not compatible with the proposed approach. In order to correctly close the transaction, the e-Node must repeat the downlink after W2 from the end of the original uplink. In **Figure 3-5** the e-Node messages have duration $M_{EU} = M_{NU}$ and $M_{ED} \neq M_{GD}$, due to possible different SF.

If Δ_U and Δ_D are the delays introduced by the e-Node in processing uplink and downlink, respectively, and Δ_{MAX} is the maximum overall time for relaying a packet, the inequalities (3.1) and (3.2) must be verified:

$$W2 - W1 = \Delta_{MAX} + M_{EU} + M_{GD} > \Delta_U + M_{EU} + \Delta_D + M_{GD} \quad (3.1)$$

$$F > \Delta_U. \quad (3.2)$$

Since W2 – W1 = 1 s, the available Δ_{MAX} can be computed as a function of the message length. In Table 3-1 and Table 3-2 Δ_{MAX} values for the join transaction, and for the maximum confirmed message length followed by an acknowledge, are reported, confirming the feasibility for the lowest SF values (i.e. for the highest data rates.

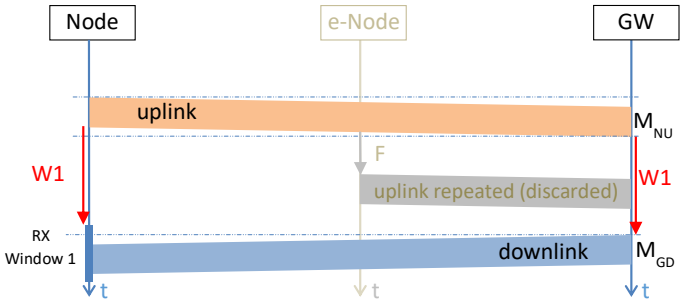


***Figure 3-4**: Time diagram of a regular uplink and downlink message exchange (for Class A devices); the e-Node is transparent to the the network.*

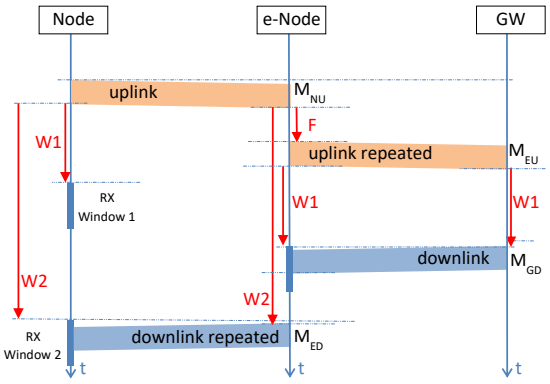


***Figure 3-5:** Time diagram of uplink and downlink message exchange when the e-Node is active in the network (for Class A devices*

Table 3-1: *feasibility of the transparent e-node for join procedure*

Data rate *SF* (CR=4/5)	**M_{EU} (ms) Join Request**	**M_{GD} (ms) Join Accept**	**Δ_{MAX} (ms)**
SF7	62	72	866
SF8	113	134	753
SF9	206	247	547
SF10	371	453	176
SF11	823	905	--
SF12	1483	1810	--

Table 3-2*: feasibility of the transparent e-node for uplink*

Data rate *SF* (CR=4/5)	**Uplink MAX PHY payload (B)**	**$M_{EU,MAX}$ (ms)**	**M_{GD} (ms) Empty ACK**	**Δ_{MAX} (ms)**
SF7	255	400	41	559
SF8	255	707	72	221
SF9	128	677	144	179
SF10	64	698	289	13
SF11	64	1561	577	--
SF12	64	2793	991	--

3.6 The LoRaWAN Range Extender Node prototype implementation

An e-Node proof-of-concept prototype has been realized to verify the feasibility and affordability of the proposed approach. The heart of the e-Node is the SX1301-SX1257 chipset from Semtech. These devices are typically embedded into LoRaWAN gateways RF frontends. The chipset only implements a regular LoRa transmitter, but also provides simultaneous reception of messages sent with different SFs on different channels. For sake of simplicity, a couple of purposely modified gateways have been used as wireless frontends. They are based on the Laird RG186-M2 daughter card (hosting the RF chipset), each one complemented by a RaspberryPi 2 running the "basic packet forwarder" from Semtech. The "basic packet forwarder" is the software in charge of tunneling LoRa messages within a backhaul network, typically a switched Ethernet. The "basic packet forwarder" establishes a connectionless UDP/IP link for forwarding JSON objects describing the actual LoRaWAN messages. Notably, received messages are encapsulated into an RxPK JSON object, whereas transmitted ones are encapsulated into a TxPK JSON object. Interesting to note, all the RxPK JSON objects have a *tmst* item, which is defined as "the value of the [SX1301] internal time counter at the instant the LoRa frame was received, with microsecond granularity". Similarly, there is a *tmst* item in the TxPk JSON objects providing "the value of the [SX1301] internal time counter at

the instant the frame will be sent, with microsecond granularity" (for TxPK). Thus, the *tmst* item can be used as a low-level timestamp for logging received messages and scheduling messages to be sent.

The "basic packet forwarder" has been modified in order to receive frames from other gateways. Indeed, LoRaWAN specifications require the inversion of in-phase and quadrature components of the baseband signal of a message sent by a node with respect to a message sent by a gateway. In this way, the gateways can only listen at messages from the nodes and vice versa.

Both the wireless frontends have been connected to a processing unit executing Node-RED flows for managing uplink and downlink messages. Node-RED is a web-based tool implemented in JavaScript and exploiting the Node.js framework. Thus, it takes advantage of Node.js built in event model, and it offers native support for JavaScript on both the client editor and the server. The obvious advantage is portability on different hardware platforms. In this secntion, two different processing units have been considered. The first one is a PC hosting an AMD Athlon X2 5600, 4 GB RAM and running a Debian 9 Linux distribution (kernel release 4.9). The second one is an IOT2040 from Siemens, featuring the energy-saving Intel Quark x1020 (+Secure Boot), 1 GB RAM, 2 Ethernet ports, 2 x RS232/485 interfaces, and one RTC; the operating system is Yocto Linux (kernel release 2.2.1).

Two separate Node-RED flows have been implemented in order to manage the JSON objects for uplink and downlink. The **Figure 3-6** shows how the e-Node prototype actually works. After an uplink transmission is demodulated, it is encapsulated in a RxPK(U) JSON object by the receiving section and the *tmst* item is stored as $tmst_{RX}(U)$. A new TxPK(U) JSON object is created, containing the received payload, and scheduled for future transmission at $tmst_{TX}(U) = tmst_{RX}(U) + F$, where F is a fixed delay sized according to the computational time of the processing unit (notably, F = 150 ms for the PC, while F = 205 ms for the IOT2040). A similar path is implemented for the downlink. After the RxPK(D) JSON object related to the message sent by the NS is received, a new TxPK(D) JSON object is created, containing the received payload, and scheduled for future transmission at $tmst_{TX}(D) = tmst_{RX}(U) + W2$. The *tmst* is obtained from the free running counter of the SX1301, consequently TxPK(U) and TxPK(D) must be sent by the same wireless frontend receiving RxPK(U). As required by the LoRaWAN specification, the channel and SF values are fixed (869.525 MHz and SF12, respectively, compliant with EU parameters).

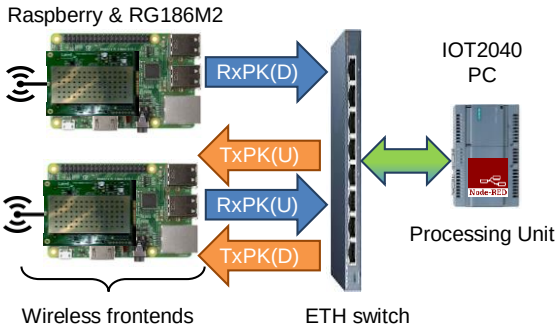


Figure 3-6: *LoRaWAN Rage Extender implementation. RxPK and TxPK refer to the packet forwarder implementation.*

The Node-RED flow executed by the processing unit for uplink messages is shown in **Figure 3-7**; a similar flow exists for the downlink.

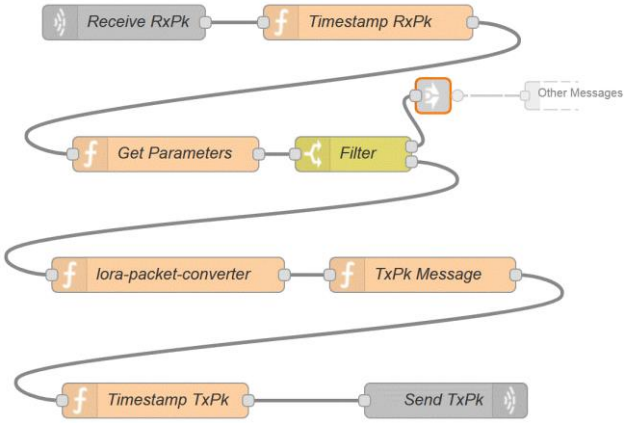


Figure 3-7: *Node-Red flow for the handling of the LoRaWAN uplink messages. (Note: The flow for the downlink direction is almost identical).*

The "Timestamp" blocks are placed immediately after and just before the native UDP "Receive" and "Send" input and output blocks. The "Get Parameter" and the "Filter" nodes select the RxPK(U) message. Subsequently, the "lora-packet-converter" node identifies LoRaWAN frame parameters available as plain text; finally, the TxPK(U) is transmitted via the "Send TxPk" node.

3.7 Experimental proof-of-concept

An experimental setup has been realized in order to verify performance of the proof-of-concept e-Node, as shown in **Figure 3-8**. In particular, a LoRaWAN compliant mote based on a Microchip RN2483 is used to send join_request, unconfirmed or confirmed messages, respectively. The RN2483 is connected to a monitoring station through an USB cable and operates as a modem. The join_requests have been transmitted using SF7 and CR=4/5; the same parameters are used for the join_accept in the first receiving window. Regular messages have a 84 B PHYPayload and have been transmitted using SF $\in$ [7,8,9] and CR=4/5. The

subsequent acknowledge for confirmed messages has variable duration; the minimum PHYPayload is 12 B, but can increase to 17 B in presence of MAC commands. The LoRaWAN network infrastructure is composed by a RG186 gateway (from Laird) connected to a PC running the LoRaServer suite, an open-source, ready-to-use LoRaWAN implementation, as shown in **Figure 3-8**.

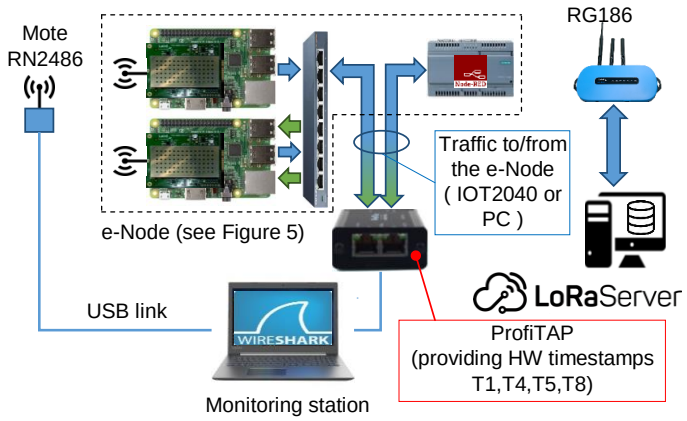


Figure 3-8: *The realized experimental setup.*

3.7.1 Coverage experiments

The verification of the increased coverage capability has been performed by means of tests carried out in the industrial laboratory located in one of the buildings of the Engineering Faculty at the University of Brescia. As shown in **Figure 3-9**, it is a typical heavy industry environment, with a surface of about 2800 m^2.

Figure 3-9: *View of the industrial laboratory at the University of Brescia where the coverage experiments have been carried out.*

The LoRaWAN Motes, based on the RN2483 modem, have been located in four different places: on the bridge crane at one end of the laboratory; in two different places on the shop floor (ground level); and in the laboratory underground basement, at -5 m level, where auxiliary machines are installed. The e-Node is just 10 m out of the laboratory at ground level,

whereas the gateway is at the first floor of the same building (about 100 m away from the e-Node).

In Table 3-3, the signal to noise ratio (SNR) and the signal strength indication (RSSI) relative to the packets directly received from the LoRaWAN Motes by the e-Node and by the gateway GW are given. The reported values are the average of 30 frames (each one 90 B long), consecutively sent every minute using SF7, SF8 and SF9, respectively. The RSSI and the SNR of the frames relayed by the e-Node and received by the gateway are not reported in the table because they are stably included within to the following ranges: RSSI= [-96, -98] dBm, SNR= [2.2 , 3.9] dB. (Note: the e-Node is at fixed position).

The results of Table 3-3 demonstrate that the use of the e-Node extends the coverage range: the Mote in the basement can be received only by the e-Node that, successively, relays the packet to the gateway. In other words, without the e-Node the basement cannot be covered by the standard LoRaWAN infrastructure. On the contrary, with the e-Node, the Mote in the basement can also operate with the fastest SFs.

Generally speaking, it can be seen that the use of the e-Node always results in an increased SNR, even when the linear distances between the Mote and the e-Node and between the Mote and the gateway are similar (as in Bridge crane position). Last, from Table 3-3 also emerges that the use of the e-Node can produce a gain (up to 16 dB) of the RSSI, especially with the fastest datarate (i.e. using SF7).

Table 3-3: *SNR and RSSI in industrial scenario.*

Mote position	**SF**	**RSSI e-Node (dBm)**	**SNR e -Node (dB)**	**RSSI GW (dBm)**	**SNR GW (dB)**
Bridge crane	SF7	-94	7.4	-91	6.3
	SF8	-95	6.5	-93	6.1
	SF9	-97	7.2	-96	5.6
Shop floor 1	SF7	-74	8.8	-90	6.7
	SF8	-76	9.5	-89	8.3
	SF9	-79	9.4	-93	8.3
Shop floor 2	SF7	-89	7.6	-96	3.7
	SF8	-90	8.6	-97	3.5
	SF9	-92	8.7	-98	4.5
Basement	SF7	-102	-0.4	missing	missing
	SF8	-103	-0.4	missing	missing
	SF9	-104	-1.7	missing	missing

3.7.2 Feasibility experiments

The most interesting feature of the proposed e-Node is the transparent behavior with respect to an already existing network. In order to meet this requirements, the time deadlines imposed by the LoRaWAN specifications must be respected; in particular the actual e-Node behavior must resemble the one depicted in **Figure 3-5**. As a consequence, time performance is of primary importance. In the following subsection, some metrics for characterizing the e-Node are proposed. Subsequently, actual experiments are described and results are discussed.

3.7.2.1 Definition of the time performance metrics

Timestamps related to the reception and the transmission of messages are collected in the Node-RED flows (as depicted in **Figure 3-7**). Timestamp on the Ethernet link (as shown in **Figure 3-8**) connecting the processing unit with the receiving and transmitting frontends, are collected using an Ethernet TAP (from ProfiTap) connected via USB to the monitoring station running Wireshark. In particular: the UDP message containing the RxPK(U) is received at T1; the uplink message is decoded by the Node-RED flow at T2; the replicated uplink is produced by the Node-RED flow at T3; the UDP message containing the replicated uplink TxPK(U) is sent at T4; the UDP message containing the RxPK(D) is received at T5; the downlink message is decoded by the Node-RED flow at T6; the replicated downlink is produced by the Node-RED flow at T7; and the UDP message containing the TxPK(D) is sent at T8.

Thanks to the collected timestamps, the following metrics can be computed:

- D_U = T3 – T2; the delay in the Node-RED application for completing the replicated uplink;
- $D_{U,TAP}$ = T4 – T1; the delay on the Ethernet link for completing the replicated uplink, taking into account the additional time spent in traversing the UDP/IP stack of the processing unit;
- D_D = T7 – T6; the delay in the Node-RED application for completing the replicated downlink;
- $D_{D,TAP}$ = T8 – T5; the delay on the Ethernet link for completing the replicated downlink;
- D_R = T7 – T2; the overall round trip time in the Node-RED application;
- $D_{R,TAP}$ = T8 – T1; the overall round trip time on the Ethernet link.

The relation (3.3) between metrics exists:

$$D_R = F + M_{EU} + W1 + M_{GD} + D_D \quad (3.3)$$

i.e. the D_R is related to all the delays in the software path.

Moreover, the following inequalities (3.4) must be satisfied

$$\Delta_U > D_{U,TAP} > D_U, \qquad \Delta_D > D_{D,TAP} > D_{D,} \tag{3.4}$$

since the delay increases when the message path between timestamping points gets longer. Last, the additional delay introduced by the Ethernet links with the LoRaWAN frontend units can be estimated with the equation (3.5):

$$P = D_{R,TAP} - D_R. \tag{3.5}$$

3.7.2.2 Experimental results

For the experiments carried out to verify time-related performance, the RG186 Gateway and the e-Node are placed at the same locations as in the previous tests. On the contrary, the RN2483 mote is placed at 5 m from the e-Node and it has a passive attenuator in series to its antenna. The RSSI of the mote is about -90dB at the e-Node receiver.

The results of the extensive (two week long) experimental campaign is reported in Table 3-5, considering 100 messages for each type sent with a time interval of one minute. The values of M_{EU} and M_{GD} are listed in Table 3-4.

The probability density function (pdf) estimates of D_U and D_D are plotted in **Figure 3-10**. The uplink path is always slower because of the more complex filter action. The asymmetry of few milliseconds is clearly visible only with IOT2040, while the high computational power of the PC slightly hides this behavior. The results of Table 3-5 confirm the asymmetry on both cases.

The pdf estimate of D_R for the JOIN message is shown in **Figure 3-11**, while for confirmed message CNF it is shown in **Figure 3-12**.

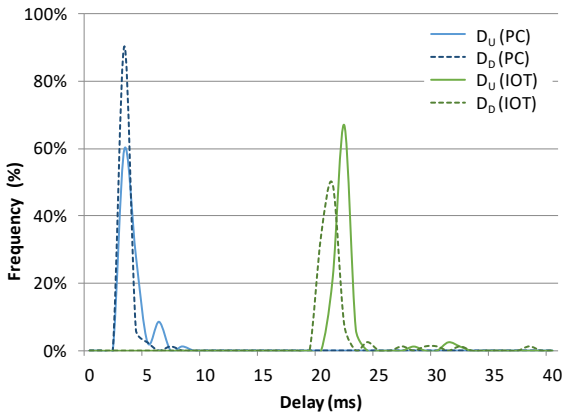


Figure 3-10: *Asymmetry of the DU and DD metrics with PC or IOT2040.*

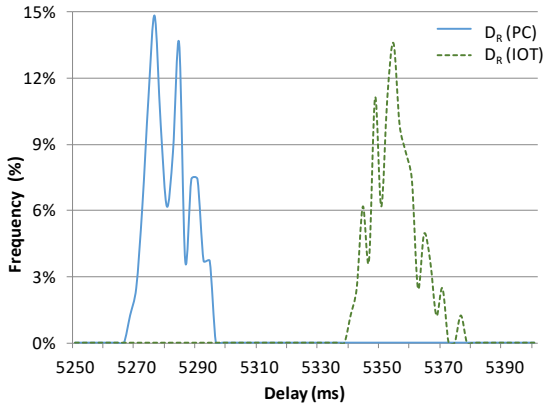


Figure 3-11: *Rountrip delay DR for JOIN messages (SF7) with PC or IOT2040.*

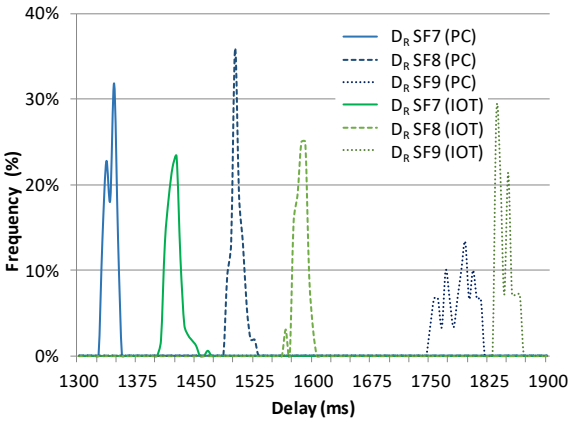


Figure 3-12: *Effect on SF on the rountrip delay DR of CNF messages.*

The join procedure is always carried out on SF7 while the CNF transmission can take place on all the three available SFs (i.e. 7, 8 and 9). The D_R metric includes the overall delay introduced by the e-Node and the M_{EU} and M_{GD} message durations. Hence, it is expected to depend on; the fixed F delay, the message length, and the SF value. The advantage of fast elaboration is visible in the graphs where, in all cases, the PC is able to forward the message with smaller delays. Calculating the difference between the D_R metrics of the two cases, and considering that the PC flow has a smaller F delay compared to IOT (150 ms for PC and 205 ms for the IOT), the advantage of the PC is about 20 ms when the minimum D_R is taken into account. This result is directly depending on the shorter computation delay D_D of the PC (see the colored cells in Table 3-5).

The delay introduced by the Ethernet links in the IOT setup can be estimated from the result of Table 3-5 using Eq. (3.5), taking into account the D_R and $D_{R,TAP}$ values (in bold in Table 3-5). The value of P metric is in the order of 30 ms, independently of the chosen SF.

Last, it should be noted that with the chosen message length it is possible to test also the SF9 messages with the slow processor of the IOT. Otherwise, the test with the SF9 would exceed the maximum delay shown in Table 3-2.

Table 3-4: *over-the-air duration of test messages.*

Message	**SF**	**Uplink length (B)**	**M_{EU} (ms)**	**Downlink length (B)**	**M_{GD} (ms)**
UNCNF	SF7	84	149	-	-
CNF	SF7	84	149	17	46
CNF	SF8	84	267	17	93
CNF	SF9	84	472	17	165
JOIN	SF7	23	62	33	72

Table 3-5*: experimental evaluation of metrics (MS).*

	Metric	**Message**	**SF**	**Min**	**Max**	**Avg**	**Std**
PC	**D_U**	UNCNF	SF7	2.0	7.0	4.0	0.9
		CNF	SF7	3.0	7.0	4.0	1.0
		CNF	SF8	3.0	6.0	3.9	0.9
		CNF	SF9	3.0	10	4.1	1.2
		JOIN	SF7	3.0	8.0	3.7	1.0
	D_D	CNF	SF7	3.0	4.0	3.4	0.6
		CNF	SF8	3.0	6.0	3.8	0.8
		CNF	SF9	3.0	10.0	4.0	1.3
		JOIN	SF7	3.0	7.0	3.2	0.6
	D_R	CNF	SF7	1330.0	1350	1338.6	6.4
		CNF	SF8	1486.0	1522.0	1500.1	7.6
		CNF	SF9	1750.0	1813.0	1783.8	19.2
		JOIN	SF7	5268.0	5294.0	5280.4	6.7
IOT	**D_U**	UNCNF	SF7	28.0	59.0	29.8	3.9
		CNF	SF7	28.0	56.0	30.1	4.0
		CNF	SF8	28.0	30.0	28.9	0.6
		CNF	SF9	28.0	39.0	31.2	4.1
		JOIN	SF7	22.0	63.0	23.2	4.8
	$D_{U,TAP}$	UNCNF	SF7	86.9	149.3	94.8	11.9
		CNF	SF7	88.7	151.1	96.7	12.1
		CNF	SF8	89.5	101.0	93.4	5.4
		CNF	SF9	86.6	147.6	96.9	12.9
		JOIN	SF7	46.3	48.5	48.5	7.3
	D_D	CNF	SF7	26.0	37.0	27.6	2.1
		CNF	SF8	26.0	28.0	26.9	0.8

		CNF	SF9	26.0	28.0	26.8	0.6
		JOIN	SF7	20.0	36.0	21.2	2.4
	$D_{D,TAP}$	CNF	SF7	58.2	100.1	62.7	7.5
		CNF	SF8	58.2	59.1	58.6	0.4
		CNF	SF9	58.9	70.7	61.3	2.8
		JOIN	SF7	43.1	45.0	45.0	4.4
	D_R	CNF	SF7	**1403.0**	1464.0	1420.0	9.5
		CNF	SF8	**1565.0**	1597.0	1582.4	7.4
		CNF	SF9	**1833.0**	1861.0	1843.4	9.0
		JOIN	SF7	**5342.0**	5382.0	5354.8	7.2
	$D_{R,TAP}$	CNF	SF7	**1435.4**	1481.1	1453.1	9.7
		CNF	SF8	**1598.0**	1623.9	1613.5	8.6
		CNF	SF9	**1867.3**	1885.4	1878.2	6.8
		JOIN	SF7	**5364.7**	5406.6	5378.8	8.0

3.8 Final Considerations for LoRa range extender

Communications for industrial applications require indoor and outdoor coverage, which pose additional issues to wireless connectivity. In particular, path loss exhibits nonlinear behavior and ensuring good link quality for all nodes can be cumbersome. When LoRaWAN is considered, the trivial approach of increasing the number of gateways could be infeasible due to cost constraints. Other currently available solutions proposed in literature are not compliant with LoRaWAN standard, thus limiting interoperability with legacy installations. In this section, the idea of an innovative relayer (the e-Node) for LoRaWAN is presented and tested. The e-Node is transparent with respect to the existing infrastructure and can be deployed to improve the link quality of poorly connected nodes, as demonstrated by experiments carried out in an industrial environment. Two kinds of e-Node hardware have been deeply tested with an extensive experimental campaign lasted for two weeks. The reported results demonstrated the usability of the e-Node, even for the higher LoRaWAN data rates typical of IIoT applications. Both join procedure and regular data exchange are correctly managed. Moreover, the experiments highlighted the need of an efficient processing unit in the e-Node to maximize the length of the messages that can be handled when the spreading factor is SF9.

Even if the e-Node is first of all a range extender, the flexibility of its hardware architecture will be further investigated in the future, in order to obtain other innovative functionalities to be applied transparently to LoRaWAN networks.

3.9 Improving LoRaWAN success probability in emergency situations

In this section, the largely diffused Low Power Area Network (LPWAN) solutions [3.23], originally designed for smart cities applications, where data from smart objects (e.g. smart meters) have to be sporadically sent to a data sink in the cloud. In particular, the focus is about the LoRaWAN technology, which has been proved to be effective and can boast a very large number of deployments. However, as aforementioned, timely message delivery and resilience are somehow overshadowed by concerns about power consumption and low-cost, limiting the use of LoRaWAN in some application fields [3.24]. To overcome these limits, this research develops an innovative message replication mechanism, called LoRa-REP, fully compliant with the LoRaWAN specifications and ensuring investment protection. In particular, in this section:

- identifies and implements two different real-world scenarios for experimental evaluation: the first one is inspired by smart city applications and it has a public and cloud-based backend; the second one is inspired by industrial situations and has local and private backend;
- reports results of a vast measurement campaign, with fixed and mobile devices based on purposely designed eNodes, aiming at evaluating obtainable performance in terms of communication resilience and timing performance.

3.10 Proposed extension of LoRaWAN for Alarm and emergency situations

In this section a brief overview of alarm management is provided in order to highlight most important requirements in terms of communication reliability and real-time behavior during emergency situations. Subsequently, the proposed LoRa-REP strategy is introduced, which permits to increase both resilience and timeliness of LoRaWAN communications. Notably, with the proposed approach, the legacy investments are preserved thanks to full compatibility with the original standard specifications.

3.10.1 Related works

The very broad adoption of LoRaWAN is testified by the interest of telecommunications providers (e.g. the Dutch telecommunications operator KPN, and The Things Network TTN), multiutility companies, and research institutes. In turn, this situation motivates the large number of works that can be found in literature. In particular, obtainable performance and limitations of the standard have been deeply analyzed. Particular interesting, in light of the strategy adopted in the proposed approach, is the idea of improving reliability by means of

replications, e.g. as suggested in [3.38][3.39], where evaluation is limited to simulations. Another important aspect is the impact that downlink traffic (including acknowledges of uplink messages) can have on the overall network throughput [3.40]-[3.42]. For this reason, downlink usage should be limited only if really required by the application. As a matter of fact, the intended IoT scenario privileges uplink, but may suffer from simultaneous activation of several nodes "sensible" to the same event, e.g. as pointed out in [3.43]; in such a case, transmission should be properly randomized, e.g. retries should happen after a certain backoff delay. Anyway, in the survey [3.41] on confirmed traffic, a more deep discussion about retry mechanism of LoRaWAN is reported. It highlights a lack in the specifications about retries, resulting in not unique interpretation so that "much of this [retry] procedure is left to the developer of the end device and NS".

3.10.2 Management of alarms and emergency

Alarm management systems are well-known in the process control industry, where they are strictly coded by international norms. The most important characteristics of alarms are: i) Prioritization, i.e. alarms should be addressed according to their criticality level; ii) Timeliness, i.e. alarms must be signaled on time; iii) Requiring response, i.e. alarms should be acknowledged in order to specify that a proper response is started.

Nowadays, alarms and acknowledgments in a plant are transferred via digital communication systems (such as fieldbuses), thus the aforementioned characteristics deeply affect requirements in terms of data exchange robustness and mixed-criticality capability, since the same medium is used for transferring information with different "importance" [3.24][3.25]. Wireless solutions have been demonstrated to be effective thanks to their flexibility [3.26]. Generally speaking, resilience is guaranteed by employing diversity and redundancy, whereas criticality is handled by means of different resource scheduling [3.27][3.28]. As a matter of fact, both needs require purposely designed MAC protocols [3.29][3.30]. Recently, alarm management in emergency conditions in complex systems (such as a whole smart city) emerged as a new research field, as demonstrated by literature [3.31][3.32].

Even if redundancy has been already used in some LoRaWAN applications, the proposed LoRa-REP replication scheme satisfies the requirements of the previous examples managing both uplink and downlink, differently from [3.33], and in a simpler way than [3.34]. Its advantages in terms of reduced delays and better success probability are deeply discussed in the following section.

3.10.3 The LoRa-REP protocol

The LoRa-REP protocol coexists with LoRaWAN and it exploits diversity and redundancy for improving the reliability of emergency communication. It must be remembered that LoRaWAN specifications define local norms about maximum transmitting power and duty-cycle. If operated in Europe, the unlicensed 868 MHz region is organized in sub-bands by the recommendation 70-03, which is normed by the ETSI standard EN300220 (requiring the duty-cycle evaluation per sub-band on 1-hour long interval). In particular, the observation interval shall be representative of the typical usage of the device, including 99% of transmissions generated during its operational lifetime. However, LoRa-REP transmissions do not contribute to the duty-cycle evaluation, since LoRa-REP-enabled nodes normally transmit following LoRaWAN rules, while a LoRa-REP transaction is generated only after a sporadic emergency event.

LoRa-REP relies on the use of different virtual channels (i.e., use of different SF values), which standard LoRaWAN has, to provide multiple redundant communication possibilities for nodes with emergency traffic. The LoRaWAN standard way of dealing with repetition of messages is not completely clear, and, as a consequence, it is rarely used. LoRa-REP has been conceived for providing a fully compliant alternative to LoRaWAN with retries. In particular, LoRa-REP considers confirmed transactions (i.e. the application requires the uplink message to be followed by a downlink confirmation) because of the emergency nature of its payload. It has to be highlighted that increasing both the communication resilience and the timeliness of confirmed transactions is of main concern. For this reason, LoRa-REP does not consider neither backoff delays between replicas nor any retransmission mechanism if all the programmed attempts fail. In case it is needed, retransmission strategy must be implemented at application level. This is a reasonable approach, since the “application” in the considered scenarios is an “emergency management” application and it may have also several other priorities to take into account.

The core idea of LoRa-REP is to replicate n times the emergency uplink frame – Uj, j=1..n – using different SFs for each replica, as displayed in **Figure 3-13**. The message copies are transmitted by “virtual nodes”, which are configured as different devices at the data link level but share the same application level. Given n confirmed uplink, also the associated n downlink confirmations – Ajk, j=1..n and k=1,2 – are expected as well (where k represents the RXk window, and it is decided by the NS). The uplink frames are transmitted from lower to higher SF values, i.e. from shorter to longer over-the-air durations. It should be noted that LoRa-REP

does not take into account the signal strength of incoming messages for optimization, i.e. the scheduling is static.

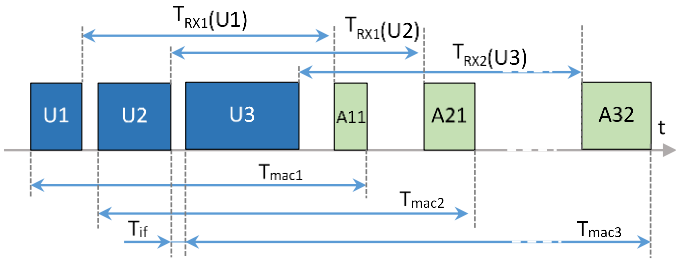


***Figure 3-13**: The LoRa-REP replication schema (example with three replicas).*

LoRa-REP is oriented to be fast and simple, hence all the Uj, j=1..n must be sent before the RX1 window of U1. Therefore, there are constraints on both the number of permitted replicas and their durations. In particular, equation (3.6) must hold, where T_{RX1} is the (fixed, but configurable) time distance between the end of an uplink message and the start of the corresponding RX1 window; T_{Uj} is the duration of the j-th uplink frame (which depends on its SF), and T_{if} is the inter-frame space between successive frames:

$$(\mathbf{n-1})\mathbf{T_{if}} + \sum_{\mathbf{j=2}}^{\mathbf{n}} \mathbf{T_{Uj}} < \mathbf{T_{RX1}} \Rightarrow \mathbf{n} < \frac{\mathbf{T_{RX1}} + \mathbf{T_{if}} - \sum_{\mathbf{j=2}}^{\mathbf{n}} \mathbf{T_{Uj}}}{\mathbf{T_{if}}}. \qquad (3.6)$$

It should be remarked that some restrictions may apply to the general case of Eq. (3.6) when constrained resource implementation of the LoRa-REP node are considered (see Section 3.11.1 for details).

As previously stated, LoRa-REP is also aimed to complete an emergency transaction (i.e. to transmit high priority message followed by a confirmation) in the least amount of time. Consequently, a significant metric is the transaction delay D_T. The minimum value $D_{T,min}$ is possible when A11 (i.e. the first RX1 downlink) is correctly received, as in (3.7), where T_{A11} is the duration of the downlink frame:

$$\mathbf{D_{T,min}} = \mathbf{T_{U1}} + \mathbf{T_{RX1}} + \mathbf{T_{A11}}. \qquad (3.7)$$

However, LoRa-REP takes advantage from higher noise immunity of higher SFs, and the transaction could be completed even if some Uj and Ajk are lost. The maximum transaction delay $D_{T,max}$ corresponds to the worst case scenario: the transaction is completed when An2 is received (i.e. the last downlink An in RX2). If T_{Ajk} is the duration of the downlink frame corresponding to the j-th uplink in the receiving window k, the expression of $D_{T,max}$ is:

$$\mathbf{D_{T,max}} = \mathbf{T_{An2}} + \mathbf{T_{RX2}} + (\mathbf{n-1})\mathbf{T_{if}} + \sum_{\mathbf{j=1}}^{\mathbf{n}} \mathbf{T_{Uj}} \qquad (3.8)$$

If the NS is configured to use only the RX1, the $D_{T,max}$ is shorter, being:

$$\mathbf{D_{T,max}} = \mathbf{T_{An1}} + \mathbf{T_{RX1}} + (\mathbf{n-1})\mathbf{T_{if}} + \sum_{\mathbf{j=1}}^{\mathbf{n}} \mathbf{T_{Uj}} \qquad (3.9)$$

There is a direct advantage over LoRaWAN standard way of dealing with retries. The time diagram of standard LoRaWAN confirmed transaction is shown in **Figure 3-14**.

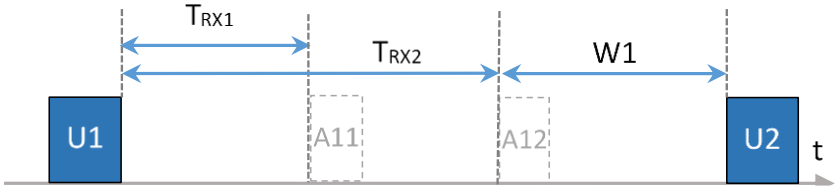


Figure 3-14: *The standard LoRaWAN replication schema.*

In LoRaWAN, the reiteration of the message transmission (on different SFs) in case of unsuccessful transaction is possible. However, the LoRaWAN specifications require that if the downlink is not received neither in RX1 nor in RX2 windows following the Uj uplink, a new uplink transmission Uj+1 can occur only after the time interval Wj. In particular, Wj must have a random contribution, $Wj_{rnd} = \mathcal{U}[-1s, +1s]$, and must be long enough to satisfy duty-cycle constraint, so that:

$$\mathbf{Wj} = \begin{cases} \mathbf{2s + Wj_{rnd}, \text{ if } T_{Uj}\left(\frac{1-DC}{DC}\right) < 2s + Wj_{rnd}} \\ \mathbf{T_{Uj}\left(\frac{1-DC}{DC}\right) + Wj_{rnd}, \text{ otherwise}} \end{cases}. \quad (3.10)$$

As a consequence, the maximum transaction delay for confirmed message in LoRaWAN, $D^{L}_{T,max}$ is:

$$\mathbf{D^{L}_{T,max} = T_{An2} + T_{RX2} + \sum_{j=1}^{n-1}(Wj + T_{RX2}) + \sum_{j=1}^{n} T_{Uj}} \quad (3.11)$$

In details, comparing (3.11) and (3.7)-(3.8), it can be observed that $\sum_{j=1}^{n-1}(Wj + T_{RX2}) \gg (n-1)T_{if}$; thus, LoRa-REP has a lower maximum transaction delay. Additionally, the minimum transaction delay is fixed in LoRa-REP and it only depends on the minimum SF chosen for the message replicas, while in LoRaWAN the minimum delay increases as the SF increases.

3.10.4 Analysis of success probability

Another important metric for LoRa-REP is the transaction success probability. Without losing generality, the considered scenario includes nodes that usually transmit unconfirmed messages (thus not generating downlink messages), while they use confirmed message (i.e. with an acknowledgment) only in case of emergency (e.g. as supposed in [3.8]). Hence, the success probability of the proposed approach can be seen as the result of the model shown in **Figure 3-15**. The success probability P_{Sj} of each parallel transaction is described by (3.12), where P_{SUj} is the probability for the GW to correctly receive the uplink Uj, and P_{SAj} is the probability for the node to correctly receive the downlink Aj:

$$\mathbf{P_{Sj} = P_{SUj} \cdot P_{SAj}.} \qquad (3.12)$$

Please note that the calculation of P_{SAj} depends on the NS strategy for transmitting the acknowledge. NS uses just one of the two downlink windows (mutually exclusive choice between RX1 or RX2). If the choice is statically configured in the NS, the probability of success is equal to the success probability of that windows: $P_{SAj} = P_{SAj1}$ or $P_{SAj} = P_{SAj2}$. Otherwise, if the choice is random $P_{SAj} = P_{RX1} \cdot P_{SAj1} + (1 - P_{RX1}) \cdot P_{SAj2}$,where the P_{RX1} is the probability of choosing RX1.

Considering that the messages transmitted using different SF values are orthogonal, and that the overall transaction is successful if at least one transaction is successful, the transaction success probability of LoRa-REP is given by:

$$\mathbf{P_S = 1 - \prod_j \left(1 - P_{SUj} \cdot P_{SAj}\right).} \qquad (3.13)$$

It is clear that, in LoRa-REP, adding single transactions to the overall transaction results in an increase of the overall success probability.

It is important to underline that the success probability of LoRaWAN with retries is the same of LoRa-REP: also the LoRaWAN retry strategy is successful if at least one transaction is successful

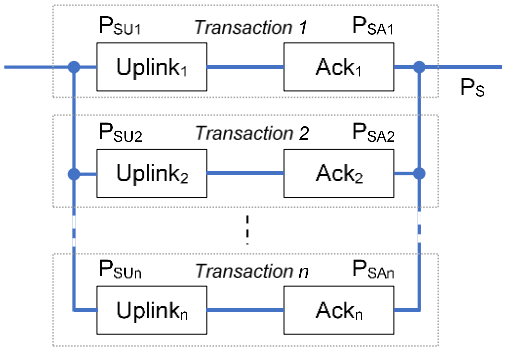


Figure 3-15: *The LoRa-REP success probability model*

3.11 Implementation of a LoRa-REP device

In this section, some considerations about the implementation of an end node with the LoRa-REP strategy are discussed. In particular, after a general description of the requirements for an engineered implementation, the prototyping platform developed by authors for speeding up on the field evaluation is introduced.

3.11.1 Mapping LoRa-REP on real devices

The LoRa-REP has been designed in order to avoid any violations of LoRaWAN, so that it can be implemented using any compliant radio. The only critical constrain is the need for Aj1 and Aj2 (j=1..n) orthogonality to ensure correct reception. Radio devices embedded into GWs permit the simultaneous reception of frames transmitted using different channels and/or SFs. Therefore, it is enough to restrict the possible uplink configuration, reducing the overall communication opportunities offered by the specifications and norms, reserving some resources for RX2 transmissions only. On the contrary, when regular end device transceivers are considered (e.g. the single chip solution SX1276 provided by Semtech), time overlapping of frames must be avoided. The easiest way to satisfy such a condition is to confine all the retries and the Aj1 messages into the T_{RX2} - T_{RX1} = 1s interval, as in (3.14)

$$(\mathbf{N}-\mathbf{1})\mathbf{T_{if}}+\sum_{\mathbf{j=2}}^{\mathbf{n}}\mathbf{T_{Uj}}+\mathbf{T_{An1}}<\mathbf{T_{RX2}}-\mathbf{T_{RX1}}. \qquad (3.14).$$

As a consequence, the number of possible replicas becomes as in (3.15)

$$\mathbf{n}<\frac{\mathbf{1}s+\mathbf{T_{if}}-\mathbf{T_{An1}}-\sum_{\mathbf{j=2}}^{\mathbf{n}}\mathbf{T_{Uj}}}{\mathbf{T_{if}}}. \qquad (3.15).$$

Another aspect to take into account is the end node lifetime, supposing a limited (e.g. from a battery) power source. In particular, focusing on the power consumption and considering the worst case scenario in which the transaction is successfully acknowledged in the first RX1 window (so that no LoRaWAN retry is actually carried out), the LoRa-REP requires n-1 additional uplink messages, each one lasting T_{Uj} j=2..n. The additional energy consumption (ΔE) can be estimated as in the following, starting from the consumption of the LoRa-REP approach (E) and the consumption of the regular LoRaWAN solution (E^L):

$$\boldsymbol{E}=\left(\sum_{\mathbf{j=1}}^{\mathbf{n}}\mathbf{T_{Uj}}\right)\cdot\boldsymbol{W_{TX}}+\mathbf{T_{A11}}\cdot\boldsymbol{W_{RX}} \qquad (3.16)$$

$$\boldsymbol{E^L}=\mathbf{T_{U1}}\cdot\boldsymbol{W_{TX}}+\mathbf{T_{A11}}\cdot\boldsymbol{W_{RX}} \qquad (3.17)$$

$$\Delta\boldsymbol{E}=\boldsymbol{E}-\boldsymbol{E^L}=\left(\sum_{\mathbf{j=2}}^{\mathbf{n}}\mathbf{T_{Uj}}\right)\cdot\boldsymbol{W_{TX}} \qquad (3.18)$$

where W_{TX} is the power consumption for transmitting and W_{RX} the power consumption for receiving. Therefore, the relative increment is:

$$\frac{\Delta\boldsymbol{E}}{\boldsymbol{E^L}}=\frac{\left(\sum_{\mathbf{j=2}}^{\mathbf{n}}\mathbf{T_{Uj}}\right)\cdot\boldsymbol{W_{TX}}}{\mathbf{T_{U1}}\cdot\boldsymbol{W_{TX}}+\mathbf{T_{A11}}\cdot\boldsymbol{W_{RX}}}. \qquad (3.19)$$

However, considering that state-of-the-art single chip transceiver draws tens of mA for receiving and transmitting messages lasting hundreds of ms, the absolute value of the additional energy ΔE required for a single LoRa-REP transaction is not significant, in the order of tens of mJ. Note that such a small ΔE value does not affect the overall device lifetime, since LoRa-REP usage is limited to emergency situations.

3.11.2 The eNode: a versatile LoRaWAN node

The feasibility of the proposed approach has been verified by means of a so called eNode, where the “e” is for enhanced. Indeed, differently from regular nodes, the core of the eNode is a radio board hosting the SX1301-SX1257 chipset from Semtech, providing baseband processor (the former) and digital-to-analog frontend (the latter). These devices are normally found in GWs (as the LoRa Gateway Board from Microchip we have actually used), and provide multiple digital receivers allowing to listen at several channels and SFs simultaneously. However, in the developed eNode, a single digital transmitter is used. The radio board is managed by a Raspberry Pi 3, which is connected via a SPI serial link. The Raspberry implements upper protocol layers; in particular, the code is based on several modules. At the bottom, there is a Hardware Abstraction Layer (HAL), derived from the “lora gateway” open source software from “LoRa-Net” repository. The HAL offers services to the “packet forwarder”, based on Semtech open source code as well.

It has to be noticed that the original “lora gateway” software has been changed to invert the in-phase and quadrature components of the baseband signal of an incoming frame, allowing the eNode to behave as an end device capable of communicating with GWs. Additionally, the original “packet forwarder”, i.e. the software responsible of injecting or retrieving wireless frames to or from the wired backhaul network, has been connected through a loopback interface to the software implementing the application level services.

In particular, the “packet forwarder” software is in charge of managing a connectionless UDP/IP link for transferring JSON objects. Indeed, RxPK JSON objects are used to encapsulate downlink messages from the backend, while TxPK JSON objects encapsulate uplink ones from the field devices. All the JSON objects have a “tmst” item, which is the value of the internal SX1301 hardware time counter when the LoRa frames are received or sent, with microsecond granularity. Therefore, the tmst item can be used for precisely timestamping messages, taking into account that the timestamping point is the end of the frame for received ones, whereas the timestamping point is the beginning of the frame for transmitted ones.

In the eNode, the “packet forwarder” sends and receives the JSON objects via the loopback connection on the port 1680 to/from the application task, which is handled by Node-RED flows. In particular, the flow-based Node-RED development tool has been chosen since it natively supports the IoT paradigm. Node-RED allows for visual programming, providing a web browser-based tool editor, which can be used to create Javascript function nodes, put into action by the underlying Node.js runtime.

Thanks to the decoupling between lower and upper sections of the protocol stack, the eNode can host several virtual nodes, all sharing the same hardware. For instance, in this section three different virtual nodes have been implemented, each one configured with static SF value (in particular, the fastest data rates permitted by SF7, SF8 and SF9 have been considered).

According to the join procedure described by the LoRaWAN specifications, each virtual node has its own unique AppKey root key and DevEUI identifier. Differing from the regular LoRaWAN join procedure, all of them share a common JoinEUI identifier. On the backend side, there is no difference at all between virtual and real nodes, so that all of them must be configured individually as usual. A flow is executed only once at the beginning for handling the Join procedure, in order to obtain the "nwkSKey" and "appSkey" session keys provided by the backend for authentication and enciphering, according to the so called OTAA procedure (normally chosen in real deployments).

During regular data exchange, each virtual node is controlled by two separate flows: one devoted to uplink and the other to downlink management, as shown in **Figure 3-16**. The function node "Get Keys VMote SFX" (where SFX stands for SF7, SF8 and SF9) returns the aforementioned "nwkSkey" and "appSkey" keys. The keys are than adopted into the "lora-packet-converter" function node, which manages LoRaWAN header parameters available as plain text, and takes care of the message ciphering and deciphering. The "TxPk Message" function node generates the uplink TxPk JSON object for the "packet forwarder". Last, the "filter" function node is in charge of filtering the downlink messages, discarding all the messages that are not intended for the eNode.

Both the uplink and downlink related flows have timestamping function nodes, that enable the tracking of when the application generates the (uplink) message and the corresponding (downlink) acknowledgment is received, retrieving the local epoch time of the Raspberry Pi.

In this section, a short overview of the reference architecture considered for experimental evaluation of the proposed LoRa-REP is sketched out and shown in **Figure 3-17**. In all the performed experiments, which took place in the same area with the same average interference, a single eNode is wirelessly connected to a regular LoRaWAN infrastructure implementing the NS and AS backend servers.

The full backward compatibility of the proposed solution has been demonstrated considering two different scenarios. In the first one, the eNode is connected to a public LoRaWAN backend infrastructure; the cloud-hosted Patavina Netsuite backend from A2A Smart City has been used in this section. In the second one, a private, local backend, based on the open source Chirpstack solution, has been considered.

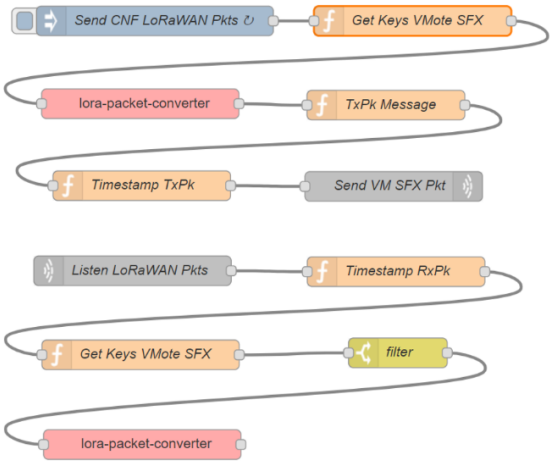


Figure 3-16: *The Node-RED flows for the uplink (top) and downlink (bottom) of the Virtual Mote (Confirmed LoRaWAN Messages) inside the eNode architecture.*

The proof-of-concept eNode device is configured to replicate application level messages by means of three virtual nodes, VNY with Y=1..3, each one exploiting a fixed SF assignment (VN1 @ SF7, VN2 @ SF8 and VN3 @ SF9, respectively; in all cases, the coding rate is CR=4/5). Transmission power is set at 14 dBm (i.e. 25 mW), which is the maximum value allowed in all the sub-bands of the unlicensed 868 MHZ band in Europe. Application level messages are generated every 60 s and require acknowledgment (i.e. they are confirmed uplink messages). Each uplink message has a 40 B user data payload; when transmitted by a virtual node, the proper 13 B-long LoRaWAN header is added. The acknowledge ACK, which is the downlink message sent by the NS, has a variable length, with a minimum raw payload of 12 B, possibly increased to 17 B if LoRaWAN MAC Commands are provided.

For sake of completeness, a specific setup for fairly comparing the LoRaWAN with retries and LoRa-REP has been added. Due to the limitations, the LoRaWAN with retries is emulated. The eNode is programmed to always transmit replicas (one for each virtual mote) of the same message using the sequence of SF derived from LoRaWAN specifications. Among all the possibilities, in the experiments of this section the number of replicas is set to Nb=3 and two sequences of SFs for the retries have been chosen in order to minimize the $D_{T,max.}$. In detail, the two sequences are: the SF7-SF7-SF7 (shortest delay) and the SF7-SF7-SF8 (longer delay but more robust). The transmission timing is equal to the LoRaWAN specifications. The outcome of each single transmission (one for each virtual mote) is recorded and, then, postprocessed. The overall transaction “fails” if all the 3 replicas do not receive their own acknowledge.

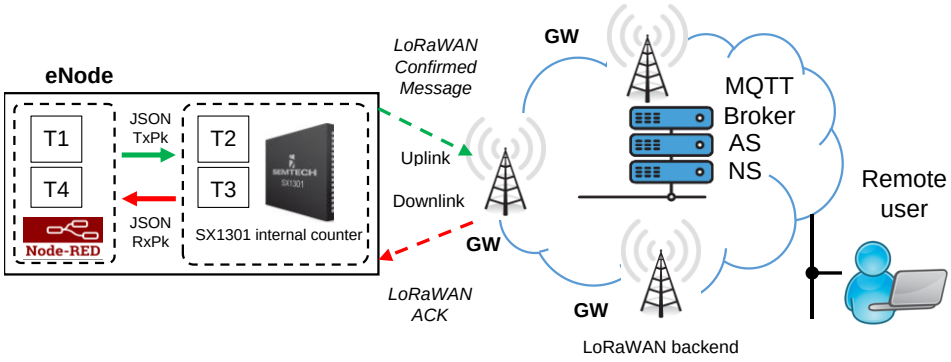


Figure 3-17: *The experimental testbed; the T1-T4 timestamps provided by the eNode are shown as well.*

3.11.3 The performance metrics

In order to assess the proposed approach performance, some metrics have been defined and evaluated. In this section, the focus is on the complete characterization of the LoRa-REP on the eNode side. Note that, conversely, evaluation of delays occurring from the on-field node to remote user has been already discussed in literature; in general, hundreds of milliseconds are required for traversing the LoRaWAN infrastructure and deliver application data to the remote user [3.36].

In particular, the timestamps inside the Node-RED flow allow the estimation of delays at the application level. The delays at the data link level are evaluated using the baseband free running internal time counter. In details, four different timestamps TK_Y (K=1..4) have been collected per each individual transaction Y=1..3. Indeed, the transaction of the virtual node VNY starts at $T1_Y$, when the JSON TxPK confirmed uplink object is sent by the Raspberry (via UDP/IP) to the packet forwarder. Each of such LoRaWAN uplink message is actually transmitted by the SX1301 baseband processor at time $T2_Y$, according to its internal counter. The GWs receiving the uplink messages forward them to the backend, where authentication is performed by the NS and the ACK downlink is automatically scheduled in the following RX1 (or RX2) window. The LoRaWAN downlink message is received by the SX1301 at $T3_Y$ (always according to its internal counter). Next, a JSON RxPK is created and sent to the Raspberry in a UDP/IP packet. Finally, the Node-RED application receives the ACK at time $T4_Y$.

In accordance with the methodology described in [3.28], some metrics have been computed for single individual uplink-downlink transaction. The single transaction delay metric D_{TY} (i.e. the time needed to conclude a transaction) is estimated as:

$$\mathbf{D_{TY} = T4_Y - T1_Y;} \quad (3.20)$$

the delay D_{IY} for traversing the LoRaWAN infrastructure (including the LoRaWAN backend servers and corresponding to the T_{macj} delay in **Figure 3-13**) can be estimated as:

$$\mathbf{D_{IY} = T3_Y - T2_Y,} \quad (3.21)$$

while the delay D_{AY} due to the application software (i.e. Node-RED in this work) can be estimated as:

$$\mathbf{D_{AY} = D_{TY} - D_{IY} = (T4_Y + T2_Y) - (T1_Y + T3_Y).} \quad (3.22)$$

Conversely, the overall transaction delay metric (i.e. the time needed to receive a confirmation from the communication partner) for the LoRa-REP is estimated as:

$$\mathbf{D_T = T4_F - T1_1,} \quad (3.23)$$

where $T4_F = \min(T4_1, T4_2, T4_3)$ is the timestamp of the first arrived downlink message in response to the LoRa-REP transaction started at $T1_1$. In other words, the LoRa-REP transaction starts using SF7 uplink and the useful downlink could be any one of the parallel transactions using SF7, SF8 or SF9. It has to be highlighted that synchronization between the Raspberry and the radio board is not required for computing the aforementioned metrics.

Finally, the effectiveness of the LoRa-REP for emergency is evaluated in terms of probability of success of the emergency transaction, i.e. when at least one of the three possible downlink is correctly received:

$$\mathbf{P_{SUCCESS} = \frac{\#\ successful\ transactions}{\#\ transactions};} \quad (3.24)$$

trivially, the loss probability is evaluated as $P_{LOSS} = 1\text{-}P_{SUCCESS}$. For the sake of comparison with regular LoRaWAN protocol applied to emergency message transmission, $P_{SUCCESS}$ is also computed for any single uplink-downlink transactionand for the emulated LoRaWAN with retry approach.

3.12 The public Cloud scenario

As previously stated, a public cloud-hosted backend has been considered for addressing typical smart city scenario. In particular, in this section the public backend is provided by the "Brescia Smart Living" (BSL) project, financed by the Italian Ministry of Education, Universities and Research, permitted to develop and test IoT-based technologies for improving the efficiency, comfort and sustainability of the Brescia municipality. One of the project partners, A2A Smart city, deployed a public LoRaWAN network covering the whole urban area (75 km^2 and 130 gateways). The backend is implemented by the Patavina Netsuite solution. The original target scenario included non-critical, non-real-time applications, as monitoring the filling level of smart bins and dumpsters or monitoring the status of the district

heater network. In such applications, tolerated update times range from few readings per day down to few readings per hour.

In this section, the authors verify the feasibility of the proposed LoRa-REP to further extend the range of applications, possibly including emergency management. In particular, experiments have been carried out in both stationary and mobile scenarios. In the former case, the node is fixed and located inside one of the open space office at the Engineering Faculty of the University of Brescia. In the latter case, the node was hosted in a car moving in the city. In all cases, tests lasted two hours (i.e. 120 confirmed messages have been sent) and the ACK downlinks were always sent in the RX1 window.

3.12.1 Experimental results with fixed node

In the experiment of this subsection the eNode is fixed indoors, whereas the public gateway closest to the eNode is outdoors (but within the University campus) at a distance of about 80 m. In order to assess the signal quality at the eNode side, the Received Signal Strength Indications (RSSI, in dBm), natively computed by the SX1301 baseband processor, is reported in Table 3-6. It must be highlighted that sensitivity better than -140 dBm can be obtained with the adopted SX1257 front-end. Additionally, the SX1301 evaluated the Signal-to-Noise Ratio (SNR, in dB) as well, which quantify the actual capability to correctly decode an incoming frame (indeed, LoRa radios are able to decode packets having SNR smaller than -15 dB, e.g. as shown in [3.37]).

Table 3-6: PUBLIC CLOUD AND FIXED NODE SCENARIO: LORA RADIO CONDITIONS DURING EXPERIMENTS

	SF7		SF8		SF9	
	SNR (dB)	RSSI (dBm)	SNR (dB)	RSSI (dBm)	SNR (dB)	RSSI (dBm)
Min	-9,5	-97,0	-10,0	-97,0	-9,8	-99,0
Ave	-6,3	-94,6	-5,9	-95,0	-5,9	-95,4
Max	-3,8	-93,0	-3,0	-92,0	-3,0	-92,0
St. dev	1,2	1,1	1,3	1,3	1,3	1,2

Thanks to the adoption of the LoRa-REP strategy, the success probability increases to 93.3% from the best, single SF transaction, which only offers 84.2% at SF9, as shown in Table 3-7. When retry strategies are used, the original emergency packet loss of LoRaWAN is more than halved, at least. On the other hand, in this scenario, the LoRa-REP seems not offering advantages in terms of success probability over LoRaWAN with retries.

Table 3-7: *PUBLIC CLOUD AND FIXED NODE SCENARIO: THE SUCCESS PROBABILITY OF LORA-REP COMPARED WITH LORAWAN*

		P_{LOSS}	$P_{SUCCESS}$
LoRaWAN	SF7 only	29.2%	70.8%
	SF8 only	20.8%	79.2%
	SF9 only	15.8%	84.2%
LoRaWAN with retries	SF7-SF7-SF7	5.0%	95.0%
	SF7-SF7-SF8	2.5%	97.5%
LoRa-REP		6.7%	93.3%

Time-related metrics are resumed in Table 3-8 and Table 3-9. As expected, the overall transaction delay D_T of the LoRa-REP reaches a minimum value corresponding to the fastest, single SF transaction and allows to lower the average value with respect to the usage of SF8 and SF9 alone, respectively. The delay D_I for traversing the LoRaWAN infrastructure is in the order of 1.3 ms, and it depends on the actual message lengths. On the contrary, the delay D_A due to the application software depends on the tasks scheduling performed by the Node-RED environment, and has an average value smaller than 200 ms, despite some (few) outliers greater than 500 ms exist.

When LoRa-REP is compared with LoRaWAN with retries (see **Table 3-9**), the clear advantage of LoRa-REP emerges. With all the considered retry strategies, LoRaWAN maximum transaction delay is several times greater than Lora-REP delay. Please note that other LoRAWAN retry sequences (not used in this section) would increase the maximum delay even more.

Table 3-8: PUBLIC CLOUD AND FIXED NODE SCENARIO: THE TRANSACTION DELAY DT OF LORA-REP AND LORAWAN

D_T [ms]		min	ave	max	Stdev
LoRaWAN	SF7 only	1313.0	1323.8	1373.0	9.1
	SF8 only	1396.5	1421.4	1766.5	50.1
	SF9 only	1610.0	1635.6	1989.0	50.1
LoRaWAN with retries	SF7-SF7-SF7	1313.0	2804.6	11076.1	2559.9
	SF7-SF7-SF8	1313.0	2875.9	11151.1	2617.8
LoRa-REP		1313.0	1430.4	2047.0	215.0

***Table 3-9**: Public Cloud and fixed node scenario: the infrastructure delay DI and application delay DA*

Metric	SF	min	ave	max	Stdev
D_I [ms]	SF7	1149.0	1149.0	1149.0	0.0
	SF8	1257.0	1257.0	1257.0	0.0
	SF9	1473.1	1473.1	1473.1	0.0
D_A [ms]	SF7	164.0	174.8	224.0	9.0
	SF8	139.5	164.4	509.5	50.0
	SF9	136.9	162.6	515.9	50.3

3.12.2 Experimental results with mobile node

In the experiment of this subsection the eNode is moved along an outdoor (city wide) path. The track is chosen in order to have good radio coverage in all locations. The test itinerary is about 20 km long and was traveled by car; the maximum speed and average speed were 50 and 15 km/h. The number of reachable gateways ranges from one to eight. Extreme, average and standard deviation values of RSSI and SNR evaluated during the whole experiment are reported in Table 3-10. It is evident that, despite minima being worse than in the fixed experiment (when the eNode, moving along the path, is more distant from the GWs), the average and maxima values are better (since some locations are closer to GWs). The slightly different radio coverage affects the success probability as well, which is slightly higher, as shown in Table 3-11. However, it is worth to notice that in both cases, the LoRa-REP is able to increase the overall success probability close to 94%, confirming advantages of the proposed approach.

Also, the LoRaWAN with retries can increase the success probability, but less than LoRA-REP. The advantage of LoRa-REP in mobile scenario can be explained by the shadowing effect of urban landscape than can block for several seconds the low SF transactions (with lower processing gain and sensitivity) used by LoRaWAN with retries. Additionally, mobile scenario is mainly outdoors, thus it is differently affected by multipath effect than fixed indoor scenario. In that situations LoRa-REP is still capable of successfully carry out the transaction exploiting the SF9 increased range and increased processing gain.

The overall transaction delay D_T is resumed in Table 3-12. As expected, there are no relevant differences with respect to the fixed scenario (indeed, infrastructure delay D_I and application delay D_A range in the same intervals as before and are not reported). Comparing LoRa-REP with LoRaWAN with retries gives indications similar to fixed scenario: the large advantage of LoRa-REP is confirmed. Interesting to note, the average transaction delay for

LoRaWAN with retries in mobile scenario is lower than in the considered fixed scenario. This can be explained because in mobile scenario more transactions complete at the first attempt (i.e. they do not require a retry) with respect to the fixed scenario.

Table 3-10: *PUBLIC CLOUD AND MOBILE NODE SCENARIO: LORA RADIO CONDITIONS DURING EXPERIMENTS*

	SF7		SF8		SF9	
	SNR (dB)	RSSI (dBm)	SNR (dB)	RSSI (dBm)	SNR (dB)	RSSI (dBm)
Min	-11,8	-120,0	-11,0	-120,0	-14,5	-120,0
Ave	2,7	-106,8	1,5	-110,0	-0,2	-111,7
Max	8,8	-71,0	8,8	-60,0	8,5	-72,0
St. dev	5,2	11,4	5,4	10,7	6,0	10,3

Table 3-11: *PUBLIC CLOUD AND MOBILE NODE SCENARIO: THE SUCCESS PROBABILITY OF LORA-REP COMPARED WITH LORAWAN*

		P_{LOSS}	$P_{SUCCESS}$
LoRaWAN	SF7 only	20.3%	79.7%
	SF8 only	17.2%	82.8%
	SF9 only	13.3%	86.7%
LoRaWAN with retries	SF7-SF7-SF7	11.7%	88.3%
	SF7-SF7-SF8	10.9%	89.1%
LoRa-REP		6.2%	93.8%

Table 3-12: *Public Cloud and mobile node scenario: the transaction delay DT of LoRa-REP and LoRaWAN*

D_T [ms]		min	ave	max	Stdev
LoRaWAN	SF7 only	1312.0	1326.2	1388.0	12.7
	SF8 only	1401.5	1421.2	2206.5	77.4
	SF9 only	1597.0	1625.7	1719.0	12.7
LoRaWAN with retries	SF7-SF7-SF7	1312.0	2127.8	10181.6	1870.6
	SF7-SF7-SF8	1312.0	2148.6	10709.6	1947.9
LoRa-REP		1312.0	1389.6	2371.0	180.7

3.13 The private local scenario

The most likely scenarios for installation of a private LoRaWAN infrastructures are industrial premises and large, highly crowded, buildings with dense structure.

In order to deal with such situations, a specific experimental setup has been created at the University of Brescia Campus. It has been designed in order to highlight feasibility and performance of the proposed LoRa-REP in such harsh environments. In detail, the LoRaWAN

network infrastructure is realized by a single Laird gateway (RG186) connected to a PC running the ChirpStack suite. ChirpStack (formerly known as "LoraServer") is a ready to use open-source implementation of complete LoRaWAN stack. The gateway is placed indoors at second floor (elevation 7 m) in the main building of the Engineering Campus.

All the experimental tests have been carried placing the eNode indoors in fixed position at different elevations (mobility is typically not considered in such a scenario). Three different situations have been considered, in which the building structure presents several windows on both sides, but many armored concrete walls and floors can separate the gateway from the eNode during the experiments.

For the sake of comparison, the first experiment has been done in the same open space office at the Engineering Faculty used for the experiments with public infrastructures described in the previous subsection. The second experiment has been done in the industrial laboratory of the Engineering Faculty. This laboratory is located in another building, with a surface of about $2700\ m^2$, and typical heavy industry setup. The last experiment has been done in the University entrance hall during rush hour, simulating a heavy crowded place (like an airport). In all three experiments, the eNode was placed at 1 m elevation from the floor. The linear distances between the eNode and the gateway are: i) 50 m for the open space experiment; ii) 70 m for the crowded hall experiment; iii) 40 m for the industrial site experiment. In all cases, tests lasted two hours (i.e. 120 confirmed messages have been sent) and the ACK downlinks are always sent in the RX1 window.

3.13.1 The experimental results in the open space

The results obtained in this experiment can be directly compared with the results of the fixed node scenario of the public cloud scenario. Signal quality for both uplink and downlink messages is reported in Table 3-13. A comparison with corresponding values reported in Table 3-6 highlights slightly better conditions, due to a better (closer) placement of the private gateway.

The LoRa-REP approach easily gains over the use of single SF with an overall success probability of 98.5%. In other terms, it means a reduction to about one-sixth of the original emergency packet loss when the best performing, single SF, transaction is addressed (i.e. from 8.5% using SF9 only to 1.5% using LoRa-REP), as shown in Table 3-14. Note that the better positioning of the private gateway permits performance even better than those reported in Table 3-7. Last, in this scenario, the LoRa-REP has a small advantage over LoRaWAN with retries, probably due to the use of SF9 for one of the three replicas. The LoRa-REP protocol is

also able to deliver the messages with an average overall delay that is lower than the most robust single SF (SF9), as visible in Table 3-15. The great advantage of LoRa-REP over LoRaWAN with retries in terms of delay is also shown in Table 3-15. Last, as expected the application delay and infrastructure delay, shown in Table 3-16, are pretty similar to the ones of the public scenarios, because the Node-Red application and the eNode hardware are the same.

Table 3-13: *Private local scenario: LoRa radio conditions during experiments*

	SF7		SF8		SF9	
	SNR (dB)	RSSI (dBm)	SNR (dB)	RSSI (dBm)	SNR (dB)	RSSI (dBm)
Min	-11,8	-107,0	-11,5	-105,0	-12,0	-105,0
Ave	-1,3	-99,8	-1,4	-100,2	-1,6	-100,3
Max	5,8	-92,0	5,5	-91,0	5,8	-91,0
St. dev	3,4	2,5	3,6	2,3	4,0	2,5

Table 3-14: *Private local scenario: the success probability of LoRa-REP compared with LoRaWAN*

		P_{LOSS}	$P_{SUCCESS}$
LoRaWAN	SF7 only	19.5%	80.5%
	SF8 only	11.5%	88.5%
	SF9 only	8.5%	91.5%
LoRaWAN with retries	SF7-SF7-SF7	7.5%	92.5%
	SF7-SF7-SF8	4.5%	95.5%
LoRa-REP		1.5%	98.5%

Table 3-15: *Private local scenario: the transaction delay DT of LoRa-REP and LoRaWAN*

D_T [ms]		min	ave	max	Stdev
LoRaWAN	SF7 only	1310.0	1321.4	1391.0	10.0
	SF8 only	1406.0	1425.5	1825.5	39.1
	SF9 only	1597.0	1624.8	1950.0	38.3
LoRaWAN with retries	SF7-SF7-SF7	1310.0	2236.4	10314.8	2138.1
	SF7-SF7-SF8	1310.0	2374.6	10978.8	2411.2
LoRa-REP		1310.0	1406.6	2360.0	206.9

Table 3-16: Private local scenario: the infrastructure delay DI and application delay DA

Metric	SF	min	ave	max	Stdev
D_I [ms]	SF7	1143.0	1143.0	1143.0	0.0
	SF8	1267.2	1267.2	1267.2	0.0
	SF9	1473.0	1473.1	1473.1	0.0
D_A [ms]	SF7	166.2	177.5	247.2	10.03
	SF8	139.3	158.3	558.3	39.1
	SF9	123.9	151.7	476.9	38.2

3.13.2 Other experiments and discussion

The results obtained in the last two experiments in terms of success/loss probabilities are reported in Table 3-17. Again the LoRa-REP was able to deliver more emergency packets than using a single SF. The gain in this case was of about 2 times for the crowded hall, while it is very low for the industrial building. The reason of the high success ratio of single SF in the industrial building is explained by: huge free space inside this site (there are no partition walls); the reduced linear distance between the eNode and the gateway; and the presence of large windows (easily traversed by the wireless signal) on the sides of both main building and industrial laboratory building.

Overall, the results obtained in the private local scenario further confirm the advantages of the propose LoRa-REP protocol, when it is applied transparently to different LoRaWAN infrastructure. As matter of fact, the underlying LoRaWAN backend (cloud or local) does not affect the LoRa-REP behavior, provided that it is able to process incoming confirmed uplink within the LoRaWAN downlink windows RX1 or RX2.

***Table 3-17**: Private local scenario: the success probability of LoRa-REP compared with LoRaWAN in different scenarios*

		Crowded Hall		Industrial building	
		P_{LOSS}	$P_{SUCCESS}$	P_{LOSS}	$P_{SUCCESS}$
LoRaWAN	SF7 only	16.5%	83.5%	2%	98%
	SF8 only	7.5%	93.5%	1%	99%
	SF9 only	4.5%	95.5%	0.5%	99.5%
LoRa-REP		2.5%	97.5%	0.5%	99.5%

3.14 Final Considerations for LoRa-REP experiments

The vast measurement campaign, carried out in different, real-world scenarios, highlighted the compatibility with already in place deployments in smart city, industrial and building automation applications, thus preserving investments. The results of the real use cases clearly

prove that the proposed approach is effective in reducing the loss of emergency messages of at least 2 times (and up to 13 times) with respect to LoRaWAN without retries. Moreover, average transaction time is reduced by 200 ms in the worst use cases. On the other hand, if LoRa-REP is compared with LoRaWAN with retries, its advantages are: a higher success probability in case of scenarios with highly (and rapidly) changing conditions (e.g. mobile); a more general reduction of the average delay (up to 1400 ms in the worst use cases); and a dramatic reduction of the maximum delay (e.g. 5 times less).

3.15 Final Considerations for Enhancing LoRaWAN Architecture

This chapter aims to mitigate some weak points inside LoRaWAN Architecture. First, related to its coverage when considering industrial applications, a repeater (the e-Node) for LoRaWAN is included and tested. This device is transparent with respect of the existing LoRaWAN infrastructure and can be deployed to improve the link quality of poorly connected nodes. The measurement campaigns demonstrated the feasibility of the proposal prototype, even in the highest peak LoRaWAN data rates commonly used in Industrial applications.

Furthermore, using the same device from the LoRa Extender, the e-Node, a LoRa-REP access method was proposed to enhance LoRaWAN technology in emergency situations. The experiments performed in different scenarios, demonstrated the compatibility with already LoRaWAN architecture. The results highlighted the proposed mechanism is efficient in increasing the feasibility of LoRaWAN transmissions in a critical situation.

Enhancing LoRaWAN technology is the last step to integrate the promise LoRaWAN solution with health applications.

4 LoRaWAN based system for Health Application

This chapter presents the arrival point of the research. After a deep investigation and enhancements in LoRaWAN technology, this chapter aims to evaluate the LoRaWAN architecture integrated with health applications for specific on-field testing in an Internet of Things scenario.

4.1 Objectives

According to the United Nations' projection, Italy is among the first three countries with the highest perspective of old-age dependency ratio in 2050 [4.1]. Ensuring assisted living care to seniors could be done in a hospice or nursing home, where health monitoring and medication management are tasks carried out by proper personnel. However, such a solution is too much expensive for most families and can become cumbersome when the elderly refuse the idea of living away from home. A recent survey from the American Association of Retired Persons (AARP) has confirmed that most of people older than 65 prefer to stay at home, even if they require continuous assistance and health care [4.2]. Indeed, it is well-known that living at home within the familiar community ensures well-being. Additionally, leaving elderly at home alone can be a great concern if the caretakers are away during the daytime.

Several research activities are ongoing for developing ambient assisted living (AAL) technologies permitting the elderly to easily look for help in the case of emergency [4.3]. AAL takes advantages from the Internet of Things (IoT) paradigm, which enables smart nodes to connect with the Cloud. In particular, Low Power Wide Area Networks (LPWANs) provide an effective answer to the request of wireless connectivity over large areas, when communication latencies comparable with human activity is tolerated. The LoRaWAN standard emerged as the most promising and versatile solution and it is currently largely adopted in many Smart Cities contexts e.g., for smart metering, since it offers the possibility to implement both public and private communication infrastructures.

In this chapter, we propose to enhance the capability of a LoRaWAN device for implementing an elderly-friendly system platform able to monitor the position and motion type of the elderly, both outdoor and indoor. The node consists of a wearable device equipped with a motion sensor (accelerometer within an inertial measuring unit or IMU) and localization engines based on GNSSs (Global Navigation Satellite Systems), operating outdoor, and UWB (Ultra Wide Band), operating indoor, respectively. The motion sensor is used to identify possible falls, so that the user's position can be broadcast to the LoRaWAN backend servers. Thus, this information can be accessed remotely in a secure way and an automatic help requests can be triggered.

According to the best authors' knowledge, it is the first time that UWB localization systems are jointly used with a LoRaWAN infrastructure for elderly assistance. Results confirm the feasibility of the proposed approach, ensuring sub-meter indoor localization error and

communication latencies on the order of few seconds. In order to understand better the chapter 4 workflow, the **Figure 4-1** shows its flowchart.

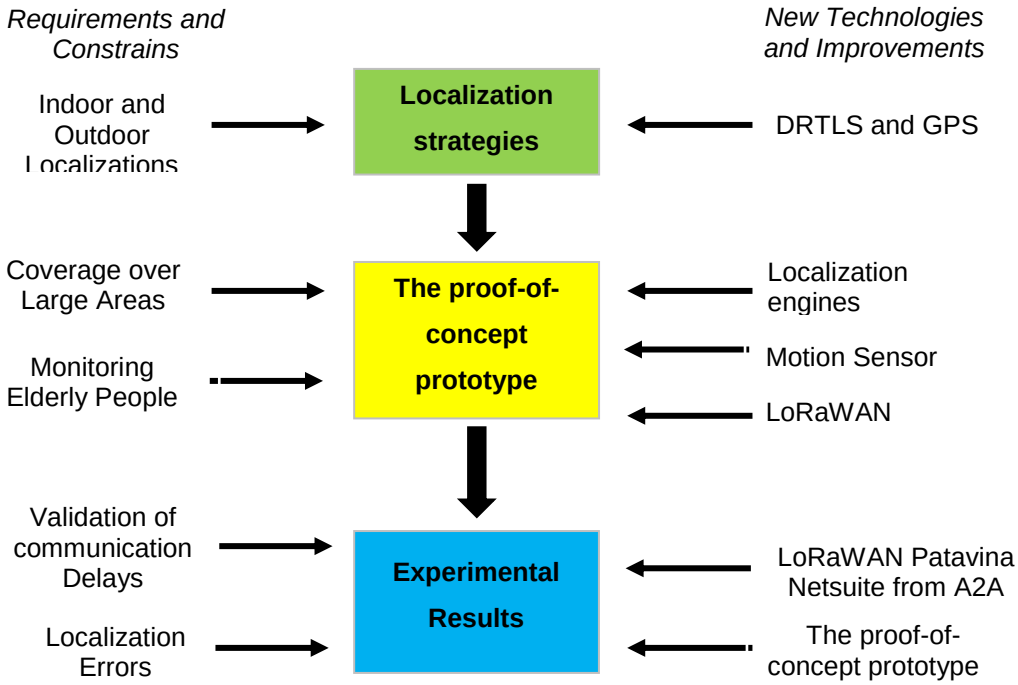


***Figure 4-1**: Graphical abstract of Chapter 4*

4.2 LoRaWAN for e-Health applications

The different LPWAN networks share some common features [4.21]:

- use sub-GHz unlicensed bands, thus avoiding the usual (and crowded) 2.4 GHz region; • offer limited throughput and channel bandwidth, permitting a very good receiver sensitivity (link budget of 150 dB can be easily obtained);
- offer low-power consumption, leveraging on minimal protocol stack complexity (on the wireless side) and supporting a star-of-star topology;
- permit low-cost end device, since all the complexity is moved into the backend system.

LoRaWAN is nowadays the most common LPWAN technology in both academia and industry. Intentionally created for IoT-related applications demanding wide coverage and sporadic transmissions, LoRaWAN networks have been deployed in many Smart Cities pilot tests. LoRaWAN nodes are based on proprietary LoRa radios from Semtech, whereas 978-1-7281-5386-5/20/$31.00 ©2020 IEEE This full text paper was peer-reviewed at the direction of IEEE Instrumentation and Measurement Society prior to the acceptance and publication. Authorized licensed use limited to: Università degli Studi di Brescia. Downloaded on January

 the LoRa Alliance, a non-profit association embracing developers and manufacturers, is in charge of managing the upper layers defined in the LoRaWAN standard.

However, the availability of open-source backend implementation makes LoRaWAN attractive for e-health applications as well, as confirmed by many available researches. In [4.22], a comprehensive research is reported about difficulties of long-term elderly healthcare in Japan; the use of IoT services, possibly leveraging on LoRaWAN, is suggested as a useful tool for solving these challenges. Other researches focus on the LoRaWAN communication performance, as in [4.23], which discusses a LoRaWAN simulator targeting urban scenarios in presence of e-health applications. In [4.24], a daily activity recognition solution is proposed as an improvement of the quality of life for elderly people in a smart healthcare center. The research in [4.25] describes an innovative smart wireless paging sensor network (WPSN) exploiting LoRaWAN and tailored for smart elderly care system. The flexibility of the LoRaWAN infrastructure, which supports both public and private implementations, is confirmed by [4.26], that describes how bio-physical parameters can be sent to an analysis center via a LoRaWAN network infrastructure. The work proposed in [4.27] discusses an advanced wearable temperature sensor, using the LoRaWAN network, to predict possible heart failures in high-risk patients.

4.3 Localization strategies

Localization techniques are normally divided into range-free and range-based solutions; in particular, in the latter case, some specific hardware is used for evaluating the device distance from reference location(s). Despite simpler, range-free solutions offer poor accuracy and are suitable only when relatively large localization errors are tolerated. Generally, reliable and precise distance measurements leverage on time-of-flight estimation.

Outdoor positioning is dominated by satellite-based solutions, grouped under the Global Navigation Satellite Systems (GNSS) umbrella, including the Global Positioning System (GPS). On the contrary, very diverse techniques and methodologies have been proposed in the past for indoor localization, exploiting Bluetooth 2.53, Radio-Frequency Identification 2.52, ZigBee [4.10], Wi-Fi [4.6]-[4.9], ultrasonic [4.11], magnetic [4.12], and even based on LED lighting [4.13].

4.3.1 Outdoor localization

GNSS solutions calculate distance by means of time-of-flight estimation and use the multilateration technique for finding the position. Obviously, time accuracy is a main concern. Atomic clocks hosted in the satellites are stable, synchronized one with each other and corrected by ground-based control stations. Clocks hosted inside end-nodes are implemented by means of crystal oscillators, because short-term stability is the only concern. When turned on, they have to synchronize with satellite clocks as well, requiring a fixing procedure which can last several minutes (depending on the fix procedure - cold, warm or hot - time to compute the distance ranges from less than 1 s to about 12 minutes). Disturbances caused by unknown satellite positions, speed of light variations in the troposphere and ionosphere and signal degradation result in positioning errors on the order of few meters [4.16].

4.3.2 Indoor localization

Indoor localization is still a difficult task and an open research topic [4.17]. Indoor localization is generally carried out by real-time locating systems (RTLSs), e.g., described by the ISO/IEC 24730 standard. Some nodes in fixed and well-known positions have the role of satellites in GNSS; generally, they are named anchors. The distance between a node (called tag) and several reference anchors can be estimated using different techniques, despite the most promising is time-of-flight estimation (as for GNSS, but operating in unlicensed bands). Subsequently, multilateration can be adopted for actual position estimation.

The most commonly adopted technique is the Symmetric Double Sided Two-Way Ranging (SDS-TWR). The SDS-TWR is an asynchronous procedure, thus involved clocks do not need to be precisely synchronized, differently from GNSS approach. Indeed, time of departure (ToD) is easily evaluated and only accurate time of arrival (ToA) estimation, i.e., timestamping of incoming message, is crucial. As shown in **Figure 4-2**, anchor and tag exchange the role of the initiator and the responder. The initiator sends a message and, after a well-defined amount of time Δt_{reply}, the responder replies. The time of flight t_p is derived from the knowledge of the ToD and the ToA of messages. Since two reply times are considered in the whole procedure, the error due to local clocks drift can be easily minimized; in fact, it depends on the difference between the two reply times, a difference that can be made short and repeatable using hardwired delays.

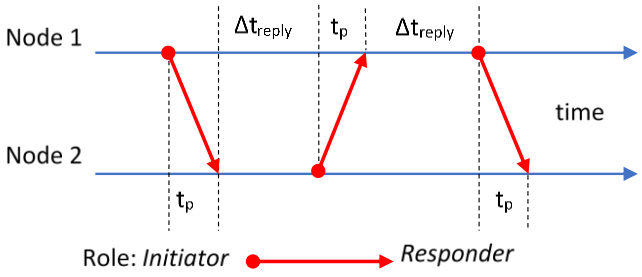


Figure 4-2: *Timing diagram showing the SDS-TWR ranging technique*

UWB communications transmitting very short pulses are most suitable for very precise ToA estimation. In this chapter, the IEEE802.15.4-UWB physical layer, used by commercially available devices as the DecaWave DWM1000, is considered. Previous works already demonstrated sub-meter errors on the position [4.18], enough for the considered application. The architecture of a DecaWave RTLS (DRTLS) is shown in **Figure 4-3**. At least three reference anchors are required for single tag localization, but no more than sixteen can operate in the same area; multi-hop topology leveraging on R nodes (for routing), not participating in the localization procedure, is allowed for disseminating messages in a broader area. According to the IEEE802.15.4 specifications, one of the anchors must behave as the DRTLS coordinator (see **Figure 4-3**), having the responsibility of starting, synchronizing and controlling the whole network. The coordinator continuously reveals its presence by transmitting Beacon messages, making possible for other anchors to join the network; once joined, an anchor can start transmitting Beacons as well.

The DWM1000 UWB transceiver is the core of both anchors and tags. The physical layer implements impulse radio UWB (IR-UWB) and works in the 6.5 GHz region of the spectrum. Each transmitted symbol is modulated by means of the burst position modulation with binary phase shift keying (BPM-BPSK), with a pulse repetition rate of 64 MHz.

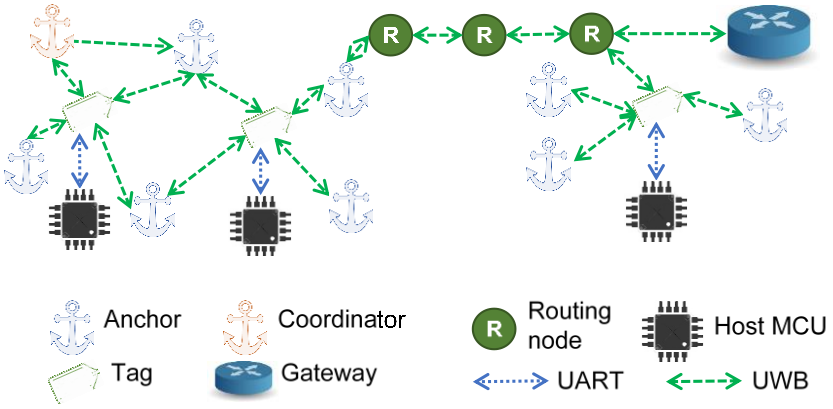


Figure 4-3: *Example of DRTLS network architecture.*

The resulting over-the-air bit rate is about 6.8 Mbps. The DRTLS medium access policy is based on Time Division Multiple Access (TDMA), thus requiring the nodes to be synchronized and to share a common sense of time. The frames exchange is structured in a "superframe" periodically repeating every 100 ms and providing communication slots for up to 15 tags. As a consequence, some scaling rules must be considered, not exceeding the maximum system localization rate of 150 node/s. In other words, it is possible to have from 15 tags localized every 100 ms (the maximum localization rate) up to 9000 tags every minute (the minimum localization rate).

Once the tag is in proximity of an area covered by enough anchors, the ranging procedure can start. Listening at the anchor messages, the tag can learn about the network topology and the anchors relative positions in a xyz-grid, so that it can estimate its own location by means of multilateration algorithm. By default, the returned position estimation is filtered by a 5-taps moving average filter and it offers a millimeter resolution.

4.4 The proof-of-concept prototype

A proof of concept prototype of the sensor device acting as a tag is shown in **Figure 4-4**. It has been developed for validating the proposed hybrid GPS and DRTLS localization solution for elderly monitoring. In particular, the system is designed around an Xplained Mini board, hosting an ATmega328P 8-bit microcontroller and supporting the well-known Arduino development environment. Intelligent modules enable connectivity and localization features. LoRaWAN Class A functionalities are ensured by a RN2483 from Microchip. GPS localization, used outdoor, is provided by means of a L80 module from Quectel, exploiting a MTK MT3339 device from Mediatek. As regards the indoor localization, the microcontroller is connected to the DWM1001 module from DecaWave, embedding the aforementioned DWM1000 radio. All the peripheral modules exploit simple UART-based communication protocols. The node is complemented by the IMU LSM9DS1 from STMicroelectronics, for acquiring acceleration signals, connected via SPI.

It is important to stress that the aim of the developed prototype is to evaluate the capability of the proposed approach to remotely notify the location of a person, as better detailed in the next section. Consequently, neither the size nor the power consumption have been optimized. Regarding the latter, power supply is provided by means of a power bank and devices are always on. The outdoor and indoor localization data are encapsulated in a LoRaWAN message and sent every minute. Fall detection evaluation and its accuracy are out of this chapter scope as well.

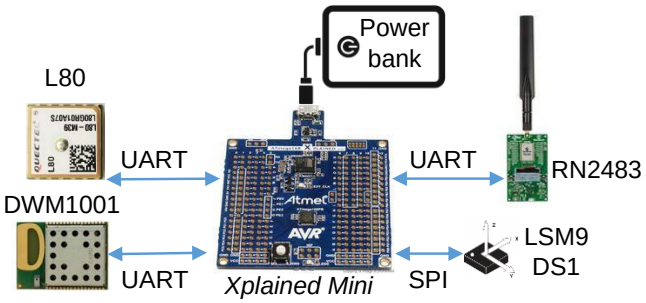


***Figure 4-4**: Intelligent peripheral modules are connected to the main microcontroller of the indoor/outdoor tag*

4.5 Experimental Results

Experimental campaigns have been performed at the University of Brescia (engineering campus), using the public LoRaWAN infrastructure of the A2A Smartcity. A2A is a multi-utility company operating in Italy. The LoRaWAN backend servers use the PTNetSuite.

4.5.1 Validation of communication delay

The experimental setup of **Figure 4-5** has been realized according to [4.14], aiming at calculating the notification delay across the public LoRAWAN infrastructure. The target performance of elderly fall detection is usually a notification delay of 1 minute, as reported in [4.15].

In the considered test scenario, there is one sensor device for the elderly people placed in the University laboratory. It transmits uplink data to a remote user endpoint (e.g., a remote PC with elderly people supervision software). The remote user is placed in another city (Milan, 80 km away from Brescia) and it uses standard ADSL connection to Internet.

The sensor device is a MQTT publisher and the remote user is a MQTT subscriber of the same topic ("event") in the MQTT Broker. The timestamp T1 is retrieved by the sensor device when a LoRaWAN uplink transmission starts. When a new message arrives in the remote user, the timestamp T3 is taken. Furthermore, the AS inside the LoRaWAN infrastructure is responsible of storing the timestamp T2. The following metrics are calculated based on these timestamps:

- the LoRaWAN infrastructure delay, i.e., ND = T2-T1;
- the MQTT broker delay, i.e., MD = T3-T2;
- and the overall end-to-end delay, i.e., OD = T3-T1.

The TM1000A NTP (+1PPS) time servers are in charge of time dissemination for both the sensor device and the remote user; they are UTC-synchronized by means of a GPS receiver. The PTNetSuite backend is natively UTC-synchronized.

The experiments have been carried in one day, with a total number of 400 LoRaWAN messages sent every 60 s with SF7 and time-on-air on the order of 230 ms. In **Figure 4-6**, the distribution of the three delay metrics is shown. The average network delay is ND = 435 ms, whereas the average MQTT Broker delay is MD = 235 ms, combined in an average overall delay OD = 680 ms. During experiments, some outliers (<3%) have been detected, reaching the maximum value of OD_{MAX} = 6250 ms.

In conclusion, the validation of the communication delay was successful: the OD is sufficiently short, making the proposed architecture suitable to be used in fall detection and notification applications in the context of elderly monitoring.

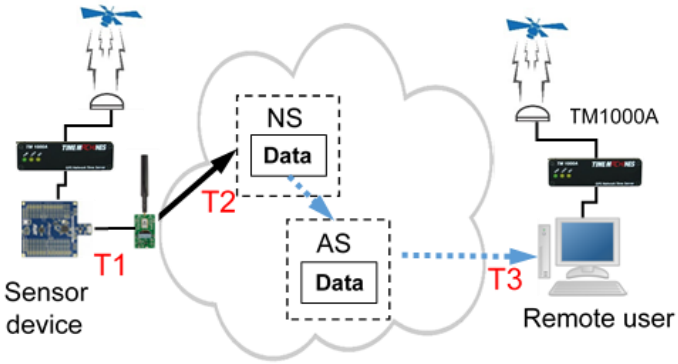


Figure 4-5: *Experimental setup with the sensor device for elderly people located 80 km away from the remote user.*

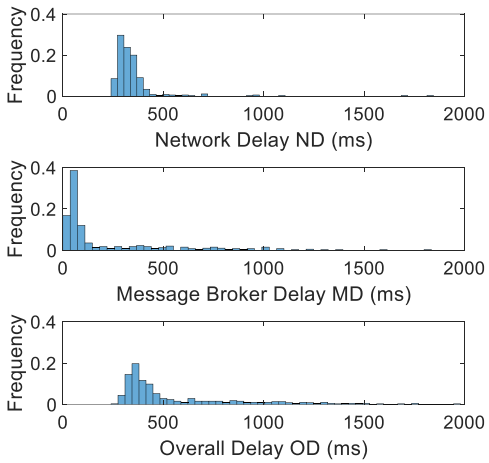


Figure 4-6: *Frequency distributions of the delay metrics in the considered scenario*

4.5.2 Localization errors

The outdoor GPS positioning accuracy has not been evaluated in this chapter, because several works investigated such topic in the past [4.16]-[4.19]. The localization test has been carried out using the following experimental setup:

- four indoor reference nodes (anchors A0, A1, A2 and A4) have been placed on vertexes of a square with side of 4000 mm (i.e., area of 16 m^2) inside a room with several obstacles;
- the tag and the anchors are hold 400 mm above the floor (mimicking a person laying on the floor), with a fully planar (2D) arrangement;
- anchor A0 is placed at coordinates (0,0) mm of the experiment area; anchor A1 is along the x-axis (4000,0) mm; anchor A2 is along the y-axis (0, -4000) mm; anchor A3 is placed at coordinates (4000,-4000) mm;
- 400 incoming data from one fixed tag (i.e., the sensor device) are stored as "estimation" of the tag position;
- three location scenarios have been tested.

The target performance of elderly fall detection system is usually on the same order of the height of a person, that is lower than 1 m [4.20]. The results of the indoor localization estimation carried out by the DecaWave RTLS are reported in Table 4-1 in terms of average values.

The distribution of the location samples for the Scenario 3 are shown in **Figure 4-7**. The standard deviation of the measurements is quite low (in the order of 15 mm) but the DRTLS system has a systematic error up to 80 mm. It should be noted that the results are taken in a real environment where multipath can affect the measurement, especially because of the reduced height from the floor.

In conclusion, the localization performance of the proposed architecture satisfies the requirements of the considered application, even without any calibration procedure.

***Table 4-1**: Localization errors for indoor scenarios (Average values, 400 samples).*

(mm)	*Tag position*		*DRTLS estimate*		*Error*	
	x	***y***	***x***	***y***	***x***	***y***
Scenario 1	550	-1100	487	-1173	87	73
Scenario 2	1600	-200	1595	-193	5	7
Scenario 3	2300	-500	2214	-530	86	30

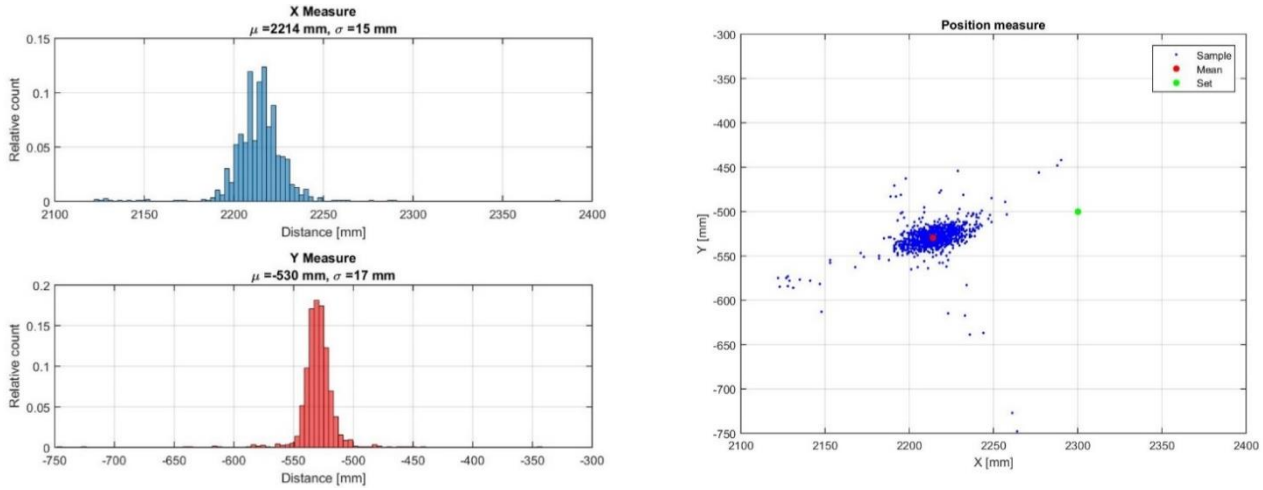


Figure 4-7: *Example of distribution of the indoor localization error for Scenario 3.*

4.6 Final Considerations for localizations services for elderly care applications with LoRaWAN

The ageing of the population poses new challenges for long term elderly care. A possible solution is the use of AAL and IoT technologies for supporting at home assistance and enable fast remote caregiver notifications.

In this chapter, the use of the most diffused LPWAN solution, i.e., LoRaWAN, has been complemented by an innovative RTLS solution, i.e., DecaWave, for implementing an elderly-friendly system platform. It is able to monitor the position and motion type of the elderly, both outdoor and indoor.

The adoption of LoRaWAN communication infrastructure permits a single technology to ensure connectivity for both indoor and outdoor scenarios, so that tracking of the end user is not limited to its own home (consider for instance a Smart City already covered by a LoRaWAN infrastructure).

The use of smart, low-cost peripheral modules speeded up the development of a proof-of-concept prototype, able to demonstrate the feasibility of the proposed approach in terms of communication latency and (indoor) localization error. An extensive measurement campaign demonstrated positioning error on the order of few centimeters indoor, whereas it is well known that GPS offers errors on the order of few tens of meters outdoor. On the other hand, notification of a remote caregiver has typical delay on the order of hundreds of milliseconds, even when best effort DSL Internet connectivity is considered.

Conclusions

Technology for Health is always improving, since the innate desire for a better life belongs to any person. New technologies offer opportunities that are often exploited to improve health. This is the case of the Internet of Things (IoT). IoT is an umbrella term describing large interconnected systems including sensor (and actuator) networks. Most of its applications rely on wireless communication, which ensures flexibility and scalability. Indeed, connecting ubiquitous IoT devices to their remote services is a complex task to manage, and the complexity of wireless connectivity is not helping. In such scenario, medical devices require more reliable network infrastructures to help doctors and assistant other medical professional in health context. This thesis describes the "research journey" to find, study, validate and enhance an effective communication solution to be applied to medical devices in an IoT scenario.

In Chapter 1, a Medical cyber-physical systems (MCPS) was suggested as a simple real-time solution to address peculiarities for detecting and limiting the impact of errors and wrong procedures for Neurodegenerative Disease Diagnosis. The proposed MCPS leverages on a message-oriented middleware, using AMQP and MQTT protocols, complemented by historical database for further data processing. The experimental setup has aimed to evaluate time-related metrics on both a local and remote networks. According to the measurements campaigns, the overall end-to-end delay from the TMS instrument to the remote client has an average value of about 140 ms, which rises up to about 160 ms for the opposite direction, confirming the feasibility of the proposed approach. It is good to mention, the experiments demonstrated that, when the geographical scale is considered, the cloud service provider may greatly impact on performance.The post analysis of the results highlighted that the wired cabling restricts the proposed MCPS approach, which greatly limits the overall flexibility in an IoT context, especially dealing with patients who need to be constantly monitored everywhere. In order to fulfill this demand, it was initiated the investigation of the state of art of wireless connectivity in an urban scenario. The connectivity of medical devices over large area is very important for medical devices and applications, since allow tracking patients inside and outside hospitals. The main scope of the thesis is LoRaWAN technology, which is the newest and promising solution in the LPWAN field. The chapter 2 describes the research of the state of the art of LoRaWAN network and the validation of its feasibility in real on-field tests. First, two main researches proposes a methodology for the experimental assessment of

architectural delays of LoRaWAN implementations in private (Semtech and LoRaIO (now ChirpStack) Server implementations) and public (Patavina Netsuite from A2A Services) networks. The proposed methodology in public LoRaWAN network is efficient to compare the time-delays performance between the two different architectures. Related to public LoRaWAN infrastructure, an experimental setup validates the complete remote end-to-end delay chain from the source and the final user of data. The results highlighted an average end-to-end delays vary between 400 and 700 ms, but abnormal peaks of several seconds have been registered.

Furthermore, the LoRaWAN network is integrated with localizations engines, a low-cost GPS module (outdoor) and UWB-based device (indoor), in a real on-field application tests. The results confirms the feasibility of the proposed device with localization errors of the few ten of meters and few tens of centimeters for outdoor and indoor positions. Moreover, the LoRaWAN technology is used as backup network in absence of 3G/4G connectivity. This chapter uses an AutoPI Edge device connected to a LoRaWAN modem, the results demonstrate the application weak points and design directions to integrate LoRaWAN network and smart Vehicles.

In Chapter 2, after stressing LoRaWAN architecture, there are two main weak points in its technology: network coverage and transmission delivery rate. Besides LoRaWAN range in open areas is in order of few kilometers, its coverage poses some restrictions in indoor scenario (e.g. inside hospitals, facilities and factories) and the delivery ratio of data packet is not satisfactory, frequently below 70%, which not fulfill the demands of applications in emergency situations. In order to bring light to solve these constraints, the Chapter 3 aims to enhance LoRaWAN architecture along two main researche paths:

- Extending LoRaWAN Coverage: a repeater (based on an enhanced LoRaWAN node, called e-Node) is addressed and tested to increase LoRaWAN coverage. The e-Node presents the integration in the infrastructure of industrial Internet of Things (IIoT)-enabled industrial wireless networks. In order to avoid the tradeoff of highest data rates against an increased sensitivity, a frame relay mechanism is demonstrated. The proposed solution is evaluated and the e-Node features, based on commercially available hardware, are tested. The results of measurements campaigns shows the feasibility of the proposed mechanism. The range extender, e-Node device, is transparent for legacy LoRaWAN networks as confirmed by correct relaying of uplink and downlink for different types of LoRaWAN messages (e.g. Join, Confirmed and Unconfirmed messages).

- Improving LoRaWAN feasibility in Emergency Situations: Using the same prototypal device from LoRaWAN extender, the e-Node, a LoRa-REP access method, which permits to increase both resilience and timeliness of LoRaWAN communications, was validated to enhance its performance in mixed criticality situations. Notably, with the proposed approach, the legacy investments are preserved thanks to full compatibility with the original standard specifications. In this way, the experiments using mobile and fixed nodes in public and private LoRaWAN networks proved the compatibility with already LoRaWAN architecture. The measurements campaigns showed the proposed mechanism is efficient in increasing the delivery ratio up to 99% and the average transaction time is reduced up to 1400 ms, considering the worst use cases.

The research outcomes are very promising demonstrating that LPWAN technology is an efficient answer to the demands of wireless connectivity over large areas in an IoT scenario, when communications delays comparable with human activity is acceptable.

Taking into account Italy, which is one of the countries with the highest perspective of old-age dependency ratio in 2050 [4.1], an application for remote monitoring daily activity of elderly people meets the local demands in health area. In this way, such solution solves the problem of assisted living care to seniors, which is done in a hospital or nursing home, where proper personnel are in charge of health monitoring and medication management, an expensive treatment for the most families. Moreover, in order to cover large areas, Low Power Wide Area Networks (LPWANs) is a suitable answer for remote assisted living care, when network requirements tolerate human daily activity. In Smart Cities contexts, the LoRaWAN technology is the most promising and versatile solution inside LPWAN scope, and it is currently largely adopted for smart metering.

For these reasons, chapter 4 describes a use case with the integration of the improved LoRaWAN solution with health applications. In details, a LoRaWAN enable device is used to monitor the position and motion type of the elderly. The node is equipped with: motion sensor, accelerometer within an inertial measuring unit (IMU) to fall detection, and localization engines based on Global Navigation Satellite Systems (GNSS) and Ultra Wide Band (UWD) for outdoor and indoor positions, respectively. The use of smart, low-cost peripheral modules speeded up the development of a proof-of-concept prototype that was able to demonstrate the feasibility of the proposed IoT approach in terms of communication latency and (indoor) localization error. The test results highlighted localization errors of tens of meters for outdoor and tens of centimeters for indoor positions. Related to timestamps metrics, overall end-to-end

delays for a notification of a remote caregiver have hundreds of milliseconds considering ADSL connections.

The obtained results confirm the effectiveness of the combination of Long Range WAN and IoT in case of eHealth applications. The possibility of using private infrastructure for building LoRaWAN systems is also very remarkable from the point of views of patience privacy.

In conclusion this PhD research contributed not only to the porting of existing technologies to MCPS devices in eHealth context, but also at the development of completely new features that are very important for real-time monitoring of health conditions and emergency situations.

References

1.1 Khaitan, S.K.; McCalley, J.D. Design Techniques and Applications of Cyberphysical Systems: A Survey. *IEEE Syst. J.***2015**, *9*, 350–365.

1.2 Al-Fuqaha, A.; Guizani, M.; Mohammadi, M.; Aledhari, M.; Ayyash, M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 2347–2376.

1.3 Crema, C.; Depari, A.; Flammini, A.; Sisinni, E.; Vezzoli, A.; Bellagente, P. Virtual Respiratory Rate Sensors: An Example of a Smartphone-Based Integrated and Multiparametric mHealth Gateway. *IEEE Trans. Instrum. Meas.* **2017**, *66*, 2456–2463.

1.4 Depari, A.; Flammini, A.; Sisinni, E.; Vezzoli, A. A wearable smartphone-based system for electrocardiogram acquisition. In Proceedings of the *Medical Measurements and Applications (MeMeA) 2014 IEEE International Symposium*; 11–12 June 2014; Lisbon, Portugal, pp. 1–6.

1.5 Zhang, X.; Li, J.; Liu, Y.; Zhang, Z.; Wang, Z.; Luo, D.; Zhou, X.; Zhu, M.; Salman, W.; Hu, G.; et al. Design of a Fatigue Detection System for High-Speed Trains Based on Driver Vigilance Using a Wireless Wearable EEG. *Sensors* **2017**, *17*, 486.

1.6 Zia ur Rehman, M.; Waris, A.; Gilani, S.O.; Jochumsen, M.; Niazi, I.K.; Jamil, M.; Farina, D.; Kamavuako, E.N. Multiday EMG-Based Classification of Hand Motions with Deep Learning Techniques. *Sensors* **2018**, *18*, 2497.

1.7 Ivanov, R.; Weimer, J.; Lee, I. Context-Aware Detection in Medical Cyber-Physical Systems. In Proceedings of the *International Conference on Cyber-Physical Systems (ICCPS)*, April, Porto, Portugal 2018.

1.8 Jezewski, J.; Pawlak, A.; Wróbel, J.; Horoba, K.; Penkala, P. Towards a medical cyber-physical system for home telecare of high-risk pregnancy. *IFAC-PapersOnLine* **2015**, *48*, 466–473.

1.9 Lee, I.; Sokolsky, O.; Chen, S.; Hatcliff, J.; Jee, E.; Kim, B.; King, A.L.; Mullen-Fortino, M.; Park, S.; Roederer, A.; et al. Challenges and Research Directions in Medical Cyber–Physical Systems. *Proc. IEEE* **2012**, *100*, 75–90.

1.10 Kocabas, O.; Soyata, T.; Aktas, M.K. Emerging Security Mechanisms for Medical Cyber Physical Systems. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2016**, *13*, 401–416.

1.11 Dey, N.; Ashour, A.S.; Shi, F.; Fong, S.J.; Tavares, J.; Manuel, R.S. Medical cyber-physical systems: A survey. *J. Med. Syst.* **2018**, *42*, 74.

1.12 Li, T.; Cao, J.; Liang, J.; Zheng, J. Towards context-aware medical cyber-physical systems: Design methodology and a case study. *Cyber Phys. Syst.*, Vo.l48, no. 4 **2015**, pp. 5–23, doi:10.1080/23335777.2014.972686.

1.13 Padovani, A.; Benussi, A.; Cantoni, V.; Dell'Era, V.; Cotelli, M.S.; Caratozzolo, S.; Turrone, R.; Rozzini, L.; Alberici, A.; Altomare, D.; et al. Diagnosis of Mild Cognitive Impairment Due to Alzheimer's Disease with Transcranial Magnetic Stimulation. *J. Alzheimer's Dis.* **2018**, *65*, 221–230.

1.14 Crema, C.; Depari, A.; Sisinni, E.; Benussi, A.; Borroni, B.; Padovani, A. Embedded platform-based system for early detection of Alzheimer disease through transcranial magnetic stimulation. In Proceedings of the 2018 IEEE Sensors Applications Symposium (SAS), Seoul, Korea, 12-14 March, 2018, pp. 350–355.

1.15 Francis, P.T.; Palmer, A.M.; Snape, M.; Wilcock, G.K. The cholinergic hypothesis of Alzheimer's disease: A review of progress. *J. Neurol. Neurosurg. Psychiatry* **1999**, *66*, 137–147.

1.16 Levenga, J.; Krishnamurthy, P.; Rajamohamedsait, H.; Wong, H.; Franke, T.F.; Cain, P.; Sigurdsson, E.M.; Hoeffer, C.A. Tau pathology induces loss of GABAergic interneurons leading to altered synaptic plasticity and behavioral impairments. *Acta Neuropathol. Commun.* **2013**, *1*, 34.
1.17 Tokimura, H.; Di Lazzaro, V.; Tokimura, Y.; Oliviero, A.; Profice, P.; Insola, A.; Mazzone, P.; Tonali, P.; Rothwell, J.C. Short latency inhibition of human hand motor cortex by somatosensory input from the hand. *J. Physiol.* **2000**, *523*, 503–513.
1.18 Kujirai, T.; Caramia, M.D.; Rothwell, J.C.; Day, B.L.; Thompson, P.D.; Ferbert, A.; Wroe, S.; Asselman, P.; Marsden, C.D. Corticocortical inhibition in human motor cortex. *J. Physiol.* **1993**, *471*, 501–519.
1.19 Ziemann, U.; Rothwell, J.C.; Ridding, M.C. Interaction between intracortical inhibition and facilitation in human motor cortex. *J. Physiol.* **1996**, *496*, 873–881.
1.20 Benussi, A.; Di Lorenzo, F.; Dell'Era, V.; Cosseddu, M.; Alberici, A.; Caratozzolo, S.; Cotelli, M.S.; Micheli, A.; Rozzini, L.; Depari, A.; et al. Transcranial magnetic stimulation distinguishes Alzheimer's disease from Frontotemporal Dementia. *Neurology* **2017**, *89*, 665–672.
1.21 Blumrosen, G.; Avisdris, N.; Kupfer, R.; Rubinsky, B. C-SMART: Efficient seamless cellular phone based patient monitoring system. In Proceedings of the 2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, Lucca, Italy, 20-24 June, 2011, pp. 1–16.
1.22 Bellagente, P.; Depari, A.; Ferrari, P.; Flammini, A.; Rinaldi, S.; Sisinni, E. M3IoT-Message-oriented middleware for M-health Internet of Things: Design and validation. In Proceedings of the 2018 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), Houston, TX, USA, 14–17 May 2018; pp. 1213–1218.
1.23 Larson, B.R.; Zhang, Y.; Barrett, S.C.; Hatcliff, J.; Jones, P.L. Enabling Safe Interoperation by Medical Device Virtual Integration. *IEEE Des. Test* **2015**, *32*, 74–88.
1.24 Plourde, J.; Arney, D.; Goldman, J.M. OpenICE: An open, interoperable platform for medical cyber-physical systems. In Proceedings of the 2014 ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS), Berlin, Germany, 14-17 April, 2014; p. 221.
1.25 Ivanov, R.; Nguyen, H.; Weimer, J.; Sokolsky, O.; Lee, I. OpenICE-lite: Towards a Connectivity Platform for the Internet of Medical Things. In Proceedings of the 2018 IEEE 21st International Symposium on Real-Time Distributed Computing (ISORC), Singapore, Singapore, 29-31 May, 2018, pp. 103–106.
1.26 Mahadevan, P.; Krioukov, D.; Fomenkov, M.; Huffaker, B.; Dimitropoulos, X.; Claffy, K.; Vahdat, A. The Internet AS-Level Topology: Three Data Sources and One Definitive Metric. *ACM SIGCOMM Comput. Commun. Rev.* **2006**, *36*, 17–26.
1.27 Tramarin, F.; Narduzzi, C.; Vitturi, S.; Bertocco, M. A Calibrated Test-Set for Measurement of Access-Point Time Specifications in Hybrid Wired/Wireless Industrial Communication. *Information* **2018**, *9*, 122.
1.28 De Vito, L.; Rapuano, S.; Tomaciello, L. One-Way Delay Measurement: State of the Art. *IEEE Trans. Instrum. Meas.* **2008**, *57*, 2742–2750.
1.29 Ferrari, P.; Sisinni, E.; Brandao, D.; Rocha, M. Evaluation of communication latency in industrial IoT applications. In Proceedings of the 2017 IEEE International Workshop on Measurements and Networking (M&N), Naples, Italy, 27–29 September 2017; pp. 17–22.
1.30 Ferrari, P.; Flammini, A.; Rinaldi, S.; Sisinni, E. Evaluation of Communication Delay in IoT Applications Based on OPC UA. In Proceedings of the 2018 Workshop on Metrology for Industry 4.0 and IoT, Brescia, Italy, 16–18 April 2018; pp. 225–230.

1.31 Fernandes Carvalho, D.; Ferrari, P.; Flammini, A.; Sisinni, E. A Test Bench for Evaluating Communication Delays in LoRaWAN Applications. In Proceedings of the 2018 Workshop on Metrology for Industry 4.0 and IoT, Brescia, Italy, 16–18 April 2018; pp. 249–254.

1.32 Silva, D.; Oliveira, G.; Silva, I.; Ferrari, P.; Sisinni, E. Latency Evaluation for MQTT and WebSocket Protocols: An Industry 4.0 Perspective. In Proceedings of the 2018 IEEE Symposium on Computers and Communications (ISCC), Natal, Brazil, 25–28 June 2018; pp. 1–6.

1.33 Ferrari, P.; Flammini, A.; Rinaldi, S.; Sisinni, E.; Maffei, D.; Malara, M. Impact of Quality of Service on Cloud Based Industrial IoT Applications with OPC UA. *Electronics* **2018**, *7*, 109.

1.34 Ferrari, P.; Flammini, A.; Sisinni, E.; Rinaldi, S.; Brandao, D.; Rocha, M. Delay Estimation of Industrial IoT Applications Based on Messaging Protocols. *IEEE Trans. Instrum. Meas.* **2018**, *67*, 2188–2199.

1.35 Tobagi, F.A.; Markopoulou, A.P.; Karam, M.J. Is the Internet ready for VoIP?. In Proceedings of the IWDC 2002; Calcutta, India, 28-31 Dec., 2002, pp. 49–57.

1.36 Collina, M.; Bartolucci, M.; Vanelli-Coralli, A.; Corazza, G.E. Internet of Things application layer protocol analysis over error and delay prone links. In Proceedings of the 7th Advanced Satellite Multimedia Systems Conference and the 13th Signal Processing for Space Communications Workshop (ASMS/SPSC), Livorno, Italy, 8-10 Sept., 2014; pp. 398–404.

1.37 Govindan, K.; Azad, A.P. End-to-end service assurance in IoT MQTT-SN. In Proceedings of the 12th Annual IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas, NV, USA, 9-12 Jan., 2015; pp. 290–296.

1.38 Mijovic, S.; Shehu, E.; Buratti, C. Comparing application layer protocols for the Internet of Things via experimentation. In Proceedings of the IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI), Bologna, Italy, 7-9 Sept., 2016; pp. 1–5.

1.39 Pereira, C.; Pinto, A.; Ferreira, D.; Aguiar, A. Experimental Characterisation of Mobile IoT Application Latency. *IEEE Internet Things J.* **2017**, *PP*, 1.

1.40 Rinaldi, S.; Pasetti, M.; Sisinni, E.; Bonafini, F.; Ferrari, P.; Rizzi, M.; Flammini, A. On the Mobile Communication Requirements for the Demand-Side Management of Electric Vehicles. *Energies* **2018**, *11*, 1220.

1.41 Sherman, J.A.; Levine, J. Usage Analysis of the NIST Internet Time Service. *J. Res. Natl. Inst. Stand. Technol. Gaithersburg* **2016**, *121*, 33–46.

1.42 Angrisani, L.; Capriglione, D.; Ferrigno, L.; Miele, G. Internet Protocol Packet Delay Variation measurements in communication networks: How to evaluate measurement uncertainty? Meas. J. Int. Meas. Conf. **2013**, 46, 2099–2109.

1.43 Hossain, T.; Ahad, A.R.A.; Tazin, T.; Inuoe S. Activity Recognition by Using LoRaWAN Sensor. Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers (UbiComp 2018), Singapore, October, 2018; pp. 58-61

1.44 Catherwood, P.A.; Steele, D.; Little, M.; Mccomb, S.; Mclaughlin, J.; A Community-Based IoT Personalized Wireless Healthcare Solution Trial. *IEEE Journal of Translational Engineering in Health and Medicine*, ***2018***, 6.

1.45 Catherwood, P.A.; Rafferty, A.; Mccomb S.; Mclaughlin, J.; LPWAN Wearable Intelligent Healthcare Monitoring for Heart Failure Prevention. Proceedings of the 32nd International BCS Human Computer Interaction Conference (HCI), Swindon, UK, 4-6 July, 2018.

1.46 Mdhaffar, A.; Chaari, T.; Larbi K.; Jmail M.; Freisleben B.; IoT-based health monitoring via LoRaWAN. Proceeding of 17th International Conference on Smart Technologies (IEEE EUROCON 2017), Ohrid, Macedonia, 6-8 July, 2017.

1.47 Islam, M.S.; Islam, M.T.; Almutairi, A.L.; Beng, G.K.; Misran, N.; Amin, N.; Monitoring of the Human Body Signal through the Internet of Things (IoT) Based LoRa Wireless Network System. *J. Applied Science, **2019**, 9*, 1884.

1.48 Patel, D.W.; Pandya, S.; Koyuncu, B.; Ramani, B.; Bhaskar, S.; Ghayvat, H. NXTGeUH: LoRaWAN based NEXT Generation Ubiquitous Healthcare System for Vital Signs Monitoring & Falls Detection. IEEE Pune Section Conference, Pune, India, 30 Nov.- 2 Dec., 2018; pp. 1-8.

1.49 "How LoRaWAN can help fight COVID-19" LoRa Alliance official web site, online at http://pages.services/pages.lora-alliance.org/covid-19-lorawan-solutions; accessed on December 15, 2020.

1.50 SigFox operator list: https://partners.sigfox.com/companies/sigfoxoperator; accessed on December 1, 2020;

1.51 The LoRa alliance web site: https://www.lora-alliance.org/; accessed on December 20, 2020;

2.1 U. Raza, P. Kulkarni and M. Sooriyabandara, "Low Power Wide Area Networks: An Overview," in IEEE Communications Surveys & Tutorials, vol. 19, no. 2, pp. 855-873, Secondquarter 2017.

2.2 C. Goursaud, J.-M. Gorce, "Dedicated networks for IoT: PHY/MAC state of the art and challenges", EAI Endorsed Trans. Internet Things, vol. 1, no. 1, Oct. 2015.

2.3 Montori F., Bedogni L., Di Felice M., Bononi L., "Machine-to-machine wireless communication technologies for the Internet of Things: Taxonomy, comparison and open issues", Pervasive and Mobile Computing, 50, 2018, pp. 56-81.

2.4 Q. Song, L. Nuaymi, X. Lagrange, "Survey of radio resource management issues and proposals for energy-efficient cellular networks that will cover billions of machines", EURASIP J. Wireless Commun. Netw., vol. 2016, no. 1, pp. 140, Dec. 2016.

2.5 H. H. R. Sherazi, M. A. Imran, G. Boggia and L. A. Grieco, "Energy Harvesting in LoRaWAN: A Cost Analysis for the Industry 4.0," in IEEE Communications Letters; early access: doi: 10.1109/LCOMM.2018.2869404.

2.6 Lin, Y.-B., Lin, Y.-W., Hsiao, C.-Y., Wang, S.-Y., "Location-based IoT applications on campus: The IoTtalk approach", Pervasive and Mobile Computing, 40, 2017, pp. 660-673.

2.7 F. Van den Abeele, J. Haxhibeqiri, I. Moerman and J. Hoebeke, "Scalability Analysis of Large-Scale LoRaWAN Networks in ns-3," in IEEE Internet of Things Journal, vol. 4, no. 6, pp. 2186-2198, Dec. 2017.

2.8 F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui and T. Watteyne, "Understanding the Limits of LoRaWAN," in IEEE Communications Magazine, vol. 55, no. 9, pp. 34-40, 2017.

2.9 K. Mikhaylov, Juha Petaejaejaervi and T. Haenninen, "Analysis of Capacity and Scalability of the LoRa Low Power Wide Area Network Technology," 22th European Wireless Conference, Oulu, Finland, 2016, pp. 1-6.

2.10 D. Magrin, M. Centenaro and L. Vangelista, "Performance evaluation of LoRa networks in a smart city scenario," 2017 IEEE International Conference on Communications (ICC), Paris, 2017, pp. 1-7.

2.11 R. B. Sørensen, D. M. Kim, J. J. Nielsen and P. Popovski, "Analysis of Latency and MAC-Layer Performance for Class A LoRaWAN," in IEEE Wireless Communications Letters, vol. 6, no. 5, pp. 566-569, Oct. 2017.

2.12 M. C. Bor, U. Roedig, T. Voigt, J. M. Alonso, "Do LoRa low-power wide-area networks scale?", Proc. 19th ACM Int. Conf. Model. Anal. Simulat. Wireless Mobile Syst., pp. 59-67, 2016.
2.13 Dvornikov, P. Abramov, S. Efremov and L. Voskov, "QoS Metrics Measurement in Long Range IoT Networks," 2017 IEEE 19th Conference on Business Informatics (CBI), Thessaloniki, 2017
2.14 F. Delobel, N. El Rachkidy and A. Guitton, "Analysis of the Delay of Confirmed Downlink Frames in Class B of LoRaWAN," 2017 IEEE 85th Vehicular Technology Conference (VTC Spring), Sydney, NSW, 2017.
2.15 D. Croce, M. Gucciardo, S. Mangione, G. Santaromita and I. Tinnirello, "Impact of LoRa Imperfect Orthogonality: Analysis of Link-Level Performance," in IEEE Communications Letters, vol. 22, no. 4, pp. 796-799, April 2018.
2.16 M. Rizzi, P. Ferrari, A. Flammini and E. Sisinni, "Evaluation of the IoT LoRaWAN Solution for Distributed Measurement Applications," in IEEE Trans. on Instr. and Meas., vol. 66, no. 12, pp. 3340-3349, Dec. 2017.
2.17 P. Ferrari, A. Flammini, M. Rizzi, E. Sisinni and M. Gidlund, "On the evaluation of LoRaWAN virtual channels orthogonality for dense distributed systems," IEEE M&N 2017 IEEE Measurement and Networking, Naples, 2017, pp. 1-6.
2.18 M. Rizzi, P. Ferrari, A. Flammini, E. Sisinni and M. Gidlund, "Using LoRa for industrial wireless networks," 2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS), Trondheim, 2017, pp. 1-4.
2.19 P. Ferrari, A. Flammini, S. Rinaldi, M. Rizzi and E. Sisinni, "On the use of LPWAN for EVehicle to grid communication," 2017 AEIT International Annual Conference, Cagliari, Italy, 2017, pp. 1-6.
2.20 S. Rinaldi, M. Pasetti, E. Sisinni, F. Bonafini, P. Ferrari, M. Rizzi, A. Flammini, On the Mobile Communication Requirements for the Demand-Side Management of Electric Vehicles. Energies, 2018, 11(5), 1220.
2.21 M. Rizzi, A. Depari, P. Ferrari, A. Flammini, S. Rinaldi, E. Sisinni, "Synchronization Uncertainty Versus Power Efficiency in LoRaWAN Networks", IEEE Trans. on Instr. and Meas., early access: doi: 10.1109/TIM.2018.2859639
2.22 A. Mahmood, E. Sisinni, L. Guntupalli, R. Rondon, S. A. Hassan and M. Gidlund, "Scalability Analysis of a LoRa Network under Imperfect Orthogonality," in IEEE Trans. on Industrial Informatics, Early access doi: 10.1109/TII.2018.2864681
2.23 D. F. Carvalho, P. Ferrari, A. Flammini and E. Sisinni, "A Test Bench for Evaluating Communication Delays in LoRaWAN Applications," 2018 Workshop on Metrology for Industry 4.0 and IoT, Brescia, 2018, pp. 248-253.
2.24 Q. Song, X. Lagrange and L. Nuaymi, "Evaluation of Macro Diversity Gain in Long Range ALOHA Networks," in IEEE Communications Letters, vol. 21, no. 11, pp. 2472-2475, Nov. 2017.
2.25 J. Navarro-Ortiz, S. Sendra, P. Ameigeiras and J. M. Lopez-Soler, "Integration of LoRaWAN and 4G/5G for the Industrial Internet of Things," in IEEE Communications Magazine, vol. 56, no. 2, pp. 60-67, Feb. 2018.
2.26 R. Yasmin, J. Petäjäjärvi, K. Mikhaylov and A. Pouttu, "On the integration of LoRaWAN with the 5G test network," 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Montreal, QC, Canada, 2017, pp. 1-6.
2.27 Luvisotto, M., Tramarin, F., Vangelista, L., Vitturi, S. “On the Use of LoRaWAN for Indoor Industrial IoT Applications”, (2018) Wireless Communications and Mobile Computing, 2018

2.28 MC Vuran, A Salam, R Wong, S Irmak, “Internet of Underground Things in Precision Agriculture: Architecture and Technology Aspects”, Ad Hoc Networks, Volume 81, December 2018, Pages 160-173
2.29 A. Ouya, B. M. De Aragon, C. Bouette, G. Habault, N. Montavont and G. Z. Papadopoulos, "An efficient electric vehicle charging architecture based on LoRa communication," 2017 IEEE International Conference on Smart Grid Communications (SmartGridComm), Dresden, 2017, pp. 381-386.
2.30 X. Ouyang, O. A. Dobre, Y. L. Guan and J. Zhao, "Chirp Spread Spectrum Toward the Nyquist Signaling Rate—Orthogonality Condition and Applications," IEEE Signal Proc. Letters, vol. 24, no. 10, pp. 1488-1492, Oct. 2017.
2.31 M. Toril and V. Wille, "Optimization of the Assignment of Base Stations to Base Station Controllers in GERAN," in IEEE Communications Letters, vol. 12, no. 6, pp. 477-479, June 2008.
2.32 M. Centenaro, L. Vangelista and R. Kohno, "On the impact of downlink feedback on LoRa performance," 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Montreal, QC, 2017, pp. 1-6.
2.33 A. Pop, U. Raza, P. Kulkarni and M. Sooriyabandara, "Does Bidirectional Traffic Do More Harm Than Good in LoRaWAN Based LPWA Networks?," GLOBECOM 2017 - 2017 IEEE Global Communications Conference, Singapore, 2017, pp. 1-6.
2.34 LoRa Alliance, “LoRaWAN specification v 1.1.”, available on line: https://lora-alliance.org/resource-hub/lorawantm-specification-v11 (accessed on October 3, 2018)
2.35 G. Yang and H. Liang, "A Smart Wireless Paging Sensor Network for Elderly Care Application using LoRaWAN," in IEEE Sensors Journal, early access doi: 10.1109/JSEN.2018.2870674
2.36 A. Lozano, J. Caridad, J. F. De Paz, G. V. González, J. Bajo, “Smart waste collection system with low consumption LoRaWAN nodes and route optimization”, Sensors 2018, 18(5), 1465.
2.37 N. Bebelaar, R. C. Braggaar, C. M. Kleijwegt, R.W.E Meulmeester, G. Michailidou, N. Salheb,S. van der Spek, N. Vaissier, E. Verbree, “Monitoring urban environmental phenomena through a wireless distributed sensor network”, Smart and Sustainable Built Environment, Volume 7, Issue 1, 3 April 2018, Pages 68-79
2.38 Official web site: https://www.thethingsnetwork.org/, Accessed on October 3, 2018
2.39 Official web site: https://www.thethingsnetwork.org, Accessed on October 3, 2018
2.40 Semtech Ltd, “LoRa IoT Starter kit user guide”, available onnlin: http://semtechlorakit.blogspot.com/ (accessed on October 3, 2018)
2.41 Semtech Ltd, “LoRaWAN Network Server Demonstration: High Level Description”, Application Note ANNWS.05.3.1.W.SYS
2.42 B. C. Fargas and M. N. Petersen, "GPS-free geolocation using LoRa in low-power WANs," 2017 Global Internet of Things Summit (GIoTS), Geneva, 2017, pp. 1-6.
2.43 S. A. A'ssri, F. H. K. Zaman and S. Mubdi, "The efficient parking bay allocation and management system using LoRaWAN," 2017 IEEE 8th Control and System Graduate Research Colloquium (ICSGRC), Shah Alam, 2017, pp. 127-131.doi: 10.1109/ICSGRC.2017.8070581
2.44 Semtech Ltd, 7, available via subscription to the “LoRa Community” https://semtech.force.com/lora”.
2.45 P. Ferrari, A. Flammini, E. Sisinni, S. Rinaldi, D. Brandao, M. Rocha, “Delay estimation of Industrial IoT applications based on messagings protocolos”, accepter for publication on the IEEE Transactions on Instrumentation and Measurement.
2.46 “The LoRa Server project”, official web site: www.loraserver.io (accessed on October 3, 2018)

2.47 S. Rinaldi, D. Della Giustina, P. Ferrari, A. Flammini, E. Sisinni, "Time synchronization over heterogeneous network for smart grid application: Design and characterization of a real case", Ad Hoc Networks, Vol. 50, pp. 41-57, 2016.
2.48 J.A. Sherman, J. Levine, "Usage Analysis of the NIST Internet Time Service", Journal of Research of the National Institute of Standards and Technology; Gaithersburg Vol. 121, (2016): 33-46
2.49 "The Things Network" official web site, online at https://www.thethingsnetwork.org/docs/network/architecture.html; accessed on January 15, 2019.
2.50 A. Potsch, F. Hammer, "Towards End-to-End Latency of LoRaWAN: Experimental Analysis and IIoT Applicability", 2019 15th IEEE World Conference on Factory Communication Systems (WFCS), Sundsvall (Sweden), 2019, pp 1-4.
2.51 Project web site: http://www.bresciasmartliving.eu/, accessed on March 15, 2018.
2.52 D. F. Carvalho, A. Depari, P. Ferrari, A. Flammini, S. Rinaldi and E. Sisinni, "On the feasibility of mobile sensing and tracking applications based on LPWAN," 2018 IEEE Sensors Applications Symposium (SAS), Seoul, 2018, pp. 1-6.
2.53 Jin, G.Y., Lu, X.Y., and Parkm, M.S., 2006. An indoor localization mechanism using active RFID tag. In: Proceedings of the IEEE international conference on sensor networks, ubiquitous, and trustworthy computing, 5–7 June, Taichung. IEEE, 40–43
2.54 Feldmann, S., et al., "An indoor bluetooth-based positioning system: concept, implementation and experimental evaluation", in proc. of the international conference on wireless networks, 23–26 June 2003, Las Vegas, NV. CSREA Press, 109–113
2.55 Bahl, P. and Padmanabhan, V.N., "Radar: an in-building RF-based user location and tracking system", in: Proc. of the 19th annual joint conference of the IEEE computer and communications societies, vol. 2, 26–30 March 2000, Tel Aviv. IEEE, 775–784
2.56 Gansemer, S., Großmann, U., and Hakobyan, S., "RSSI-based Euclidean distance algorithm for indoor positioning adapted for the use in dynamically changing WLAN environments and multi-level buildings", in proc. of the 1st the international conference on indoor positioning and indoor navigation, 15–17 Sept. 2010, Zurich. IEEE, 1–6
2.57 Machaj, J., Brida, P., and PicheÌ, R., "Rank based fingerprinting algorithm for indoor positioning", in: Proc. of the 2nd international conference on indoor positioning and indoor navigation, 21–23 Sept., Guimarães, 2011.
2.58 Marques, N., Meneses, F., Moreira, A., "Combining similarity functions and majority rules for multi-building, multi-floor, Wi-Fi positioning", in: Proc. of the 3rd international conference on indoor positioning and indoor navigation, 13–15 Nov. 2012, Sydney.
2.59 Marti, J.V., et al. "Localization of mobile sensors and actuators for intervention in low-visibility conditions: the zigbee fingerprinting approach", Int. Journal of Distributed Sensor Networks, 2012, 1–10.
2.60 Ijaz, F., et al., "Indoor positioning: A review of indoor ultrasonic positioning systems", in Proc. of the 15th international conference on advanced communication technology, 27–30 Jan. 2013, PyeongChang. IEEE, 1146–1150
2.61 Chung, J., et al., 2011. Indoor location sensing using geo-magnetism. In: A.K.Agrawala, M.D. Corner, and D. Wetherall, eds. Proceedings of the 9th international conference on mobile systems, applications, and services, 28 June–1 July, Washington, DC. ACM, 141–154
2.62 Kuo, Y.S., et al., "Luxapose: indoor positioning with mobile phones and visible light", in proc. of the 20th annual international conference on mobile computing and networking, 7–11 Sept. 2014, Maui, 447–458.

2.63 Yong He, Haihong Yu and Hui Fang, "Study on Improving GPS Measurement Accuracy," 2005 IEEE Instrumentationand Measurement Technology Conference Proceedings, Ottawa, Ont., 2005, pp. 1476-1479.
2.64 A. Yassin et al., "Recent Advances in Indoor Localization: A Survey on Theoretical Approaches and Applications," in IEEE Communications Surveys & Tutorials, vol. 19, no. 2, pp. 1327-1346, Secondquarter 2017.
2.65 W. Chantaweesomboon et al., "On performance study of UWB real time locating system," 2016 7th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES), Bangkok, 2016, pp. 19-24.
2.66 M. Strohmeier, T. Walter, J. Rothe and S. Montenegro, "Ultra-Wideband Based Pose Estimation for Small Unmanned Aerial Vehicles," in IEEE Access, vol. 6, pp. 57526-57535, 2018.
2.67 M.A. Forin-Wiart, P. Hubert, P. Sirguey, M.-L. Poulle, “Performance and accuracy of lightweight and low-cost GPS data loggers according to antenna positions, fix intervals, habitats and animal movements”, (2015) PLoS ONE, 10 (6).
2.68 M. Pelka, G. Goronzy, H. Hellbrück, “Iterative approach for anchor configuration of positioning systems”, ICT Express, Vol. 2, No. 1, March 2016, Pages 1-4
2.69 M. S. Hossain, "Cloud-Supported Cyber–Physical Localization Framework for Patients Monitoring," in IEEE Systems Journal, vol. 11, no. 1, pp. 118-127, March 2017.
2.70 Silva M., Vieira E., Signoretti G., Silva I., Silva D., Ferrari P., “A Customer Feedback Platform for Vehicle Manufacturing Compliant with Industry 4.0 Vision”, Sensors (Basel). 2018; v. 18, n.10, pp. 3298. 2018
2.71 R. Malekian, N. R. Moloisane, L. Nair, B. T. Maharaj and U. A. K. Chude-Okonkwo, "Design and Implementation of a Wireless OBD II Fleet Management System," in IEEE Sensors Journal, vol. 17, no. 4, pp. 1154-1164, 15 Feb.15, 2017.
2.72 Pieroni A., Scarpato N., Brilli M. Industry 4.0 revolution in autonomous and connected vehicle a now-conventional approach to manage Big Data. J. Theor. Appl. Inf. Technol. 2018;96:10–18
2.73 Silva, M., Signoretti, G., Oliveira, J., Silva, I., Costa, Daniel, I. “A Crowdsensing Platform for Monitoring of Vehicular Emissions: A Smart City Perspective”, Smart Cities, 2, 1, 46-65, 2019.
2.74 G. Signoretti, Silva M., A. Dias, Silva I., Silva D., Ferrari P., “Performance Evaluation of an Edge OBD-II Device for Industry 4.0”, Proc. of IEEE Workshop on Metrology for Industry 4.0 and IoT, Naples, 4-6 June, 2019
2.75 Bonafini F., Fernandes Carvalho D., Depari A., Ferrari P., Flammini A., Pasetti M., Rinaldi S., Sisinni E., “Evaluating indoor and outdoor localization services for LoRaWAN in Smart City applications”, Proc. of IEEE Workshop on Metrology for Industry 4.0 and IoT, Naples, 4-6 June, 2019
2.76 D. F. Carvalho, A. Depari, P. Ferrari, A. Flammini, S. Rinaldi and E. Sisinni, "On the feasibility of mobile sensing and tracking applications based on LPWAN," 2018 IEEE Sensors Applications Symposium (SAS), Seoul, 2018, pp. 1-6.

3.1 E. Sisinni, D. F. Carvalho, P. Ferrari, A. Flammini, D. R. C. Silva and I. M. D. Da Silva, "Enhanced flexible LoRaWAN node for industrial IoT," 2018 14th IEEE International Workshop on Factory Communication Systems (WFCS), Imperia, 2018, pp. 1-4.
3.2 M. Rizzi, P. Ferrari, A. Flammini and E. Sisinni, "Evaluation of the IoT LoRaWAN Solution for Distributed Measurement Applications," in IEEE Trans. on Instrumentation and Measurement, vol. 66, no. 12, pp. 3340-3349, Dec. 2017

3.3 M. Rizzi, A. Depari, P. Ferrari, A. Flammini, S. Rinaldi and E. Sisinni, "Synchronization Uncertainty Versus Power Efficiency in LoRaWAN Networks," in IEEE Transactions on Instrumentation and Measurement", doi: 10.1109/TIM.2018.2859639

3.4 L. Tessaro, C. Raffaldi, M. Rossi and D. Brunelli, "LoRa Performance in Short Range Industrial Applications," 2018 International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM), Amalfi, 2018, pp. 1089-1094.

3.5 M. Rizzi, P. Ferrari, A. Flammini, E. Sisinni and M. Gidlund, "Using LoRa for industrial wireless networks," 2017 IEEE 13th Workshop on Factory Communication Systems (WFCS), Trondheim, 2017, pp. 1-4.

3.6 J. Haxhibeqiri, A. Karaagac, F. Van den Abeele, W. Joseph, I. Moerman and J. Hoebeke, "LoRa indoor coverage and performance in an industrial environment: Case study," 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Limassol, 2017, pp. 1-8

3.7 B. Vejlgaard, M. Lauridsen, H. Nguyen, I. Z. Kovacs, P. Mogensen and M. Sorensen, "Interference Impact on Coverage and Capacity for Low Power Wide Area IoT Networks," 2017 IEEE Wireless Communications and Networking Conference (WCNC), San Francisco, CA, 2017, pp. 1-6.

3.8 M. Luvisotto, F. Tramarin, L. Vangelista, S. Vitturi, "On the use of LoRaWAN for indoor industrial IoT applications", Wirel. Commun. Mob. Comput., 2018 (2018)

3.9 J. Navarro-Ortiz, S. Sendra, P. Ameigeiras and J. M. Lopez-Soler, "Integration of LoRaWAN and 4G/5G for the Industrial Internet of Things," in IEEE Communications Magazine, vol. 56, no. 2, pp. 60-67, Feb. 2018.

3.10 H. C. Lee and K. H. Ke, "Monitoring of Large-Area IoT Sensors Using a LoRa Wireless Mesh Network System: Design and Evaluation," in IEEE Transactions on Instrumentation and Measurement (early access).

3.11 C. H. Liao, G. Zhu, D. Kuwabara, M. Suzuki and H. Morikawa, "Multi-Hop LoRa Networks Enabled by Concurrent Transmission," in IEEE Access, vol. 5, pp. 21430-21446, 2017.

3.12 B. Sartori, S. Thielemans, M. Bezunartea, A. Braeken and K. Steenhaut, "Enabling RPL multihop communications based on LoRa,", IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Rome, 2017, pp. 1-8.

3.13 Weiwei ZhouZiyuan TongZ.Y. DongZ.Y. DongYu WangYu Wang, "LoRa-Hybrid: A LoRaWAN Based multihop solution for regional microgrid", February 2019, Conference: 2019 IEEE 4th International Conference on Computer and Communication SystemsAt: Singapore

3.14 F. Cuomo, M. Campo, A. Caponi, G. Bianchi, G. Rossini and P. Pisani, "EXPLoRa: Extending the performance of LoRa by suitable spreading factor allocations," 2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Rome, 2017, pp. 1-8.

3.15 F. Parzysz, M. Vu and F. Gagnon, "Impact of Propagation Environment on Energy-Efficient Relay Placement: Model and Performance Analysis," in IEEE Transactions on Wireless Communications, vol. 13, no. 4, pp. 2214-2228, April 2014

3.16 M. Anedda, C. Desogus, M. Murroni, D. D. Giusto and G. Muntean, "An Energy-efficient Solution for Multi-Hop Communications in Low Power Wide Area Networks," 2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), Valencia, 2018, pp. 1-5.

3.17 A. Willig and E. Uhlemann, "On relaying for wireless industrial communications: Is careful placement of relayers strictly necessary?," 2012 9th IEEE International Workshop on Factory Communication Systems, Lemgo, 2012, pp. 191-200;
3.18 A. Willig, "Placement of relayers in wireless industrial sensor networks: An approximation algorithm," 2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Singapore, 2014, pp. 1-6
3.19 N. T. Tuan, D. Kim and J. Lee, "On the Performance of Cooperative Transmission Schemes in Industrial Wireless Sensor Networks," in IEEE Trans. on Ind. Informatics, vol. 14, no. 9, pp. 4007-4018, Sept. 2018.
3.20 Z. Qin, D. Wu, Z. Xiao, B. Fu and Z. Qin, "Modeling and Analysis of Data Aggregation From Convergecast in Mobile Sensor Networks for Industrial IoT," in IEEE Trans. on Ind. Informatics, vol. 14, no. 10, pp. 4457-4467, Oct. 2018.
3.21 A. Mahmood, E. Sisinni, L. Guntupalli, R. Rondon, S. A. Hassan and M. Gidlund, "Scalability Analysis of a LoRa Network under Imperfect Orthogonality," in IEEE Trans. on Ind. Informatics; doi: 10.1109/TII.2018.2864681
3.22 Rady M., Hafeez M., Zaidi S.A.R., “Computational methods for network-aware and network-agnostic IoT low power wide area networks (LPWANs)”, IEEE Internet of Things Journal, 6 (3), pp. 5732-5744, 2019. 4
3.23 M. Pasetti, P. Ferrari, D.R.C. Silva, I. Silva, E. Sisinni, “On the use of LoRaWAN for the monitoring and control of distributed energy resources in a smart campus”, Applied Sciences (Switzerland), 10 (1), 2020 5
3.24 A. Burns, R. I. Davis, S. Baruah and I. Bate, "Robust Mixed-Criticality Systems", IEEE Transactions on Computers, vol. 67, no. 10, pp. 1478-1491, 1 Oct. 2018. 15
3.25 H. Farag, E. Sisinni, M. Gidlund and P. Österberg, "Priority-Aware Wireless Fieldbus Protocol for Mixed-Criticality Industrial Wireless Sensor Networks", IEEE Sensors Journal, vol. 19, no. 7, pp. 2767-2780, 2018. 16
3.26 Fu J.-S., Liu Y., Chao H.-C., Zhang Z.-J., “Green alarm systems driven by emergencies in industrial wireless sensor networks”, IEEE Communications Magazine, 54 (10), pp. 16-21, 2016. 17
3.27 A. Willig, "Redundancy concepts to increase transmission reliability in wireless industrial LANs," in IEEE Transactions on Industrial Informatics, vol. 1, no. 3, pp. 173-182, Aug. 2005. 18
3.28 G. Cena, S. Scanzio and A. Valenzano, "Experimental Evaluation of Seamless Redundancy Applied to Industrial Wi-Fi Networks," in IEEE Trans. on Industrial Informatics, vol. 13, no. 2, pp. 856-865, April 2017. 19
3.29 W. Shen, T. Zhang, F. Barac, M. Gidlund, "PriorityMAC: A priority-enhanced MAC protocol for critical traffic in industrial wireless sensor and actuator networks", IEEE Trans. Ind. Informat., vol. 10, no. 1, pp. 824-835, Feb. 2014. 20
3.30 L. Leonardi, F. Battaglia, L. Lo Bello, "RT-LoRa: A Medium Access Strategy to Support Real-Time Flows Over LoRa-Based Networks for Industrial IoT Applications," in IEEE Internet of Things Journal, vol. 6, no. 6, pp. 10812-10823, Dec. 2019. 21
3.31 Abhishek R., Zhao S., Tipper,D., Medhi D., “SeSAMe: Software defined smart home alert management system for smart communities”, IEEE Workshop on Local and Metropolitan Area Networks, 2017-June, 2017 22
3.32 R Bhatti F., Shah M.A.S., Maple C., Ul Islam S., “A novel internet of things-enabled accident detection and reporting system for smart city environments”, Sensors (Switzerland), 19 (9), 2019 23

3.33 S. Borkotoky, C. Bettstetter, U. Schilcher and C. Raffelsberger, “Allocation of Repetition Redundancy in LoRa”, in Proc. Of 25th European Wireless Conference, Aarhus, Denmark, 2019, pp. 1-6. 24
3.34 P. Marcelis, V. Rao, and R. V. Prasad, “DaRe: Data recovery through application layer coding for LoRaWAN” In Proc. Of ACM/IEEE Int. Conf. IoT Des. Impl., pp. 97–108, April 2017 25
3.35 Available on-line https://github.com/Lora-net/; accessed on April 2, 2020. 27
3.36 D. Fernandes Carvalho et al., "A test methodology for evaluating architectural delays of LoRaWAn implementations", Pervasive and Mobile Computing, vol. 56, no. 5, pp. 1-17, 2019. 28
3.37 P. Neumann, J. Montavont, T. Noël, "Indoor deployment of low-power wide area networks (LPWAN): A LoRaWAN case study," 2016 IEEE 12th Int. Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), New York, NY, pp. 1-8, 2016. 29
3.38 A. Hoeller, R. D. Souza, O. L. Alcaraz Lòpez, H. Alves, M. de Noronha Neto and G. Brante, "Exploiting Time Diversity of LoRa Networks Through Optimum Message Replication," 2018 15th Int. Symposium on Wireless Communication Systems (ISWCS), Lisbon, 2018, pp. 1-5.
3.39 J. M. d. S. Sant’Ana, A. Hoeller, R. D. Souza, S. Montejo-Sánchez, H. Alves and M. d. Noronha-Neto, "Hybrid Coded Replication in LoRa Networks," in IEEE Trans. on Industrial Informatics, vol. 16, no. 8, pp. 5577-5585, Aug. 2020
3.40 Goursaud C., Gorce J.M., “Dedicated networks for IoT : PHY / MAC state of the art and challenges”, EAI endorsed transactions on Internet of Things, 2015
3.41 G. Margelis, R. Piechocki, D. Kaleshi and P. Thomas, "Low Throughput Networks for the IoT: Lessons learned from industrial implementations," 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), Milan, 2015, pp. 181-186.
3.42 S. Penkov, A. Taneva, V. Kalkov and S. Ahmed, "Industrial network design using Low-Power Wide-Area Network," 2017 4th International Conference on Systems and Informatics (ICSAI), Hangzhou, 2017, pp. 40-44.
3.43 L. Leonardi, F. Battaglia, G. Patti and L. L. Bello, "Industrial LoRa: A Novel Medium Access Strategy for LoRa in Industry 4.0 Applications," IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society, Washington, DC, 2018, pp. 4141-4146

4.1 United Nations, “World Population Ageing 2019”, ST/ESA/SER.A/430, Department of Economic and Social Affairs Population Division. 1
4.2 “Most Retirees Prefer to Stay Put”, AARP Bulletin, October 2018. Available on line (accessed on Jan. 27, 2020) https://www.aarp.org/retirement/planning-for-retirement/info-2018/retirees-age-in-place-aarp-study.html. 2
4.3 V. Bianchi et al., "IoT Wearable Sensor and Deep Learning: An Integrated Approach for Personalized Human Activity Recognition in a Smart Home Environment," in IEEE Internet of Things Journal, vol. 6, no. 5, pp. 8553-8562, Oct. 2019. 3
4.4 Feldmann, S., et al., “An indoor bluetooth-based positioning system: concept, implementation and experimental evaluation”, in proc. of the international conference on wireless networks, 23–26 June 2003, Las Vegas, NV. CSREA Press, 109–113 13
4.5 Jin, G.Y., Lu, X.Y., and Parkm, M.S., 2006. An indoor localization mechanism using active RFID tag. In: Proceedings of the IEEE international conference on sensor networks, ubiquitous, and trustworthy computing, 5–7 June, Taichung. IEEE, 40–43 14

4.6 Bahl, P. and Padmanabhan, V.N., “Radar: an in-building RF-based user location and tracking system”, in: Proc. of the 19th annual joint conference of the IEEE computer and communications societies, vol. 2, 26–30 March 2000, Tel Aviv. IEEE, 775–784 15
4.7 Gansemer, S., Großmann, U., and Hakobyan, S., “RSSI-based Euclidean distance algorithm for indoor positioning adapted for the use in dynamically changing WLAN environments and multi-level buildings”, in proc. of the 1st Int. conf. on indoor positioning and indoor navigation, 15–17 Sept. 2010, Zurich. IEEE, 1–6 16
4.8 Machaj, J., Brida, P., and PicheÌ, R., “Rank based fingerprinting algorithm for indoor positioning”, in: Proc. of the 2nd international conference on indoor positioning and indoor navigation, 21–23 Sept., Guimarães, 2011. 17
4.9 Marques, N., Meneses, F., Moreira, A., “Combining similarity functions and majority rules for multi-building, multi-floor, Wi-Fi positioning”, in: Proc. of the 3rd international conference on indoor positioning and indoor navigation, 13–15 Nov. 2012, Sydney. 18
4.10 Marti, J.V., et al. “Localization of mobile sensors and actuators for intervention in low-visibility conditions: the zigbee fingerprinting approach”, Int. Journal of Distributed Sensor Networks, 2012, 1–10. 19
4.11 Ijaz, F., et al., “Indoor positioning: A review of indoor ultrasonic positioning systems”, in Proc. of the 15th international conference on advanced communication technology, 27–30 Jan. 2013, PyeongChang, 1146–1150 20
4.12 Chung, J., et al., 2011. Indoor location sensing using geo-magnetism. In: A.K.Agrawala, M.D. Corner, and D. Wetherall, eds. Proceedings of the 9th international conference on mobile systems, applications, and services, 28 June–1 July, Washington, DC. ACM, 141–154 21
4.13 Kuo, Y.S., et al., “Luxapose: indoor positioning with mobile phones and visible light”, in proc. of the 20th annual international conference on mobile computing and networking, 7–11 Sept. 2014, Maui, 447–458. 22
4.14 D. F. Carvalho, A. Depari, P. Ferrari, A. Flammini, S. Rinaldi and E. Sisinni, "On the evaluation of application level delays in public LoRaWAN networks," 2019 IEEE International Symposium on Measurements & Networking (M&N), Catania, Italy, 2019, pp. 1-6 23
4.15 C. Wang et al., "Low-Power Fall Detector Using Triaxial Accelerometry and Barometric Pressure Sensing," in IEEE Transactions on Industrial Informatics, vol. 12, no. 6, pp. 2302-2311, Dec. 2016. 24
4.16 Yong He, Haihong Yu and Hui Fang, "Study on Improving GPS Measurement Accuracy," 2005 IEEE Instrumentationand Measurement Technology Conference Proceedings, Ottawa, Ont., 2005, pp. 1476-1479. 25
4.17 A. Yassin et al., "Recent Advances in Indoor Localization: A Survey on Theoretical Approaches and Applications," in IEEE Communications Surveys & Tutorials, vol. 19, no. 2, pp. 1327-1346, Secondquarter 2017. 26
4.18 W. Chantaweesomboon et al., "On performance study of UWB real time locating system," 2016 7th Int. Conf. of Inf. and Comm. Technology for Embedded Systems (IC-ICTES), Bangkok, 2016, pp. 19-24. 27
4.19 M. Matosevic, Z. Salcic and S. Berber, "A Comparison of Accuracy Using a GPS and a Low-Cost DGPS," in IEEE Transactions on Instrumentation and Measurement, vol. 55, no. 5, pp. 1677-1683, Oct. 2006. 28
4.20 Z. Wang, Z. Yang, and T. Dong, “A Review of Wearable Technologies for Elderly Care that Can Accurately Track Indoor Position, Recognize Physical Activities and Monitor Vital Signs in Real Time,” Sensors, vol. 17, no. 2, p. 341, Feb. 2017. 31

4.21 U. Raza, P. Kulkarni and M. Sooriyabandara, "Low Power Wide Area Networks: An Overview," in IEEE Communications Surveys & Tutorials, vol. 19, no. 2, pp. 855-873, Secondquarter 2017.
4.22 Koekuza, H., “Utilization of loT in the Long-term Care Field in Japan”, in 2nd International Conference on Cloud Computing and Internet of Things (CCIOT), Dalian, China (2016)
4.23 Wang, H., Fapojuwo, A., “Performance Evaluation of LoRaWAN In North America Urban Scenario”, in IEEE 88th Vehicular Technology Conference (VTC-Fall), Chicago, USA (2018)
4.24 Hossain, T., Inoue, S.. “Sensor-based Daily Activity Understanding in Caregiving Center”, in Sensor-based Daily Activity Understanding in Caregiving Center, Kyoto, Japan (2019)
4.25 Yang, G., Liang, H., “A Smart Wireless Paging Sensor Network for Elderly Care Application Using LoRaWAN”, in IEEE Sensors Journal, Vol. 18, No. 22, November 15, 2018.
4.26 Mdhaffar, A., Chaari, T., Larbi, K., Jmaiel, M., Bernd, F.: IoT-based Health Monitoring via LoRaWAN, in IEEE EUROCON – 17th Internacional Conference on Smart Technologies, Ohrid, Macedonia, July, 2017
4.27 Catherwood, P., Rafferty, J., McComb, S., McLaughlin, J., “LPWAN Wearable Intelligent Healthcare Monitoring for Heart Failure Prevention”, in 20th Internacional Conference on Human Computer Interaction, Las Vegas, Nevada, EUA, 2018

DICHIARAZIONE DI CONFORMITÀ DELLE TESI
PER IL CONSEGUIMENTO DEL TITOLO DI DOTTORE DI RICERCA
(DICHIARAZIONE SOSTITUTIVA DI ATTO NOTORIO E DI CERTIFICAZIONE
(**artt. 46-47 del D.P.R. 445 del 28.12.00 e relative modifiche)**

Il/La sottoscritto Dhiego Fernandes Carvalho Nato/a il 26/041985
a Brasile Provincia/Stato Natal/RN
Dottorato di ricerca in Technology for Health
a conoscenza del fatto che in caso di dichiarazioni mendaci, oltre alle sanzioni previste dal Codice Penale e dalle Leggi speciali per l'ipotesi di falsità in atti ed uso di atti falsi, decade dai benefici conseguenti al provvedimento emanato sulla base di tali dichiarazioni,

DICHIARA

sotto la propria responsabilità, ai fini dell'ammissione all'esame finale per il conseguimento del titolo di Dottore di ricerca,

di essere a conoscenza che,

in conformità al Regolamento dell'Università degli Studi di Brescia per l'ammissione all'esame finale ed il rilascio del titolo per il conseguimento del titolo di Dottore di Ricerca, è tenuto a depositare all'U.O.C. Dottorati e Scuole di Specializzazione:
- n. 1 copia in formato cartaceo della propria tesi di dottorato e **l'esposizione riassuntiva (*abstract)* in italiano, se la redazione della tesi è stata autorizzata in lingua straniera;**
- n. 2 copie della tesi su DVD o CD-ROM per il deposito presso le Biblioteche Nazionali di Roma e di Firenze;

DICHIARA inoltre

- che il contenuto e l'organizzazione della tesi sono opera originale e non compromettono in alcun modo i diritti di terzi,
- **che sarà consultabile** immediatamente dopo il conseguimento del titolo di Dottore di ricerca, in quanto non è il risultato di attività rientranti nella normativa sulla proprietà industriale, non è stata prodotta nell'ambito di progetti finanziati da soggetti pubblici o privati con vincoli alla divulgazione dei risultati, e non è oggetto di eventuali registrazioni di tipo brevettale o di tutela.
- che l'Università è in ogni caso esente da qualsiasi responsabilità di qualsivoglia natura, civile, amministrativa o penale e sarà tenuta indenne da qualsiasi richiesta o rivendicazione da parte di terzi;
- che la tesi in formato elettronico (DVD o CD-ROM) è completa in ogni sua parte ed è del tutto identica a quella depositata in formato cartaceo all'U.O.C. Dottorati e Scuole di Specializzazione dell'Università degli Studi di Brescia e inviata ai Commissari. Di conseguenza va esclusa qualsiasi responsabilità dell'Ateneo per quanto riguarda eventuali errori, imprecisioni od omissioni nei contenuti della tesi.

Dichiara inoltre di essere consapevole che saranno effettuati dei controlli a campione. Eventuali discordanze od omissioni potranno comportare l'esclusione dal dottorato di ricerca;

Luogo e data Brescia/BS - 04/02/2021 Firma del dichiarante