Enabling robust wireless communication for battery management systems in electric vehicles

Fabian Rincon Vija

DOCTORAT BRETAGNE LOIRE / MATHSTIC

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

THESE DE DOCTORAT DE

L'ÉCOLE NATIONALE SUPERIEURE MINES-TELECOM ATLANTIQUE
BRETAGNE PAYS DE LA LOIRE - IMT ATLANTIQUE

ECOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Informatique*

Par
Fabian Antonio RINCON VIJA

Enabling Robust Wireless Communication for Battery Management Systems in Electric Vehicles

Thèse présentée et soutenue à IMT Atlantique Campus Rennes, le 14 décembre 2022
Unité de recherche : IRISA
Thèse N° : 2022IMTA0318

Rapporteurs avant soutenance :

Thomas WATTEYNE	Directeur de recherche, INRIA équipe AIO Paris
Carlo Alberto BOANO	Professeur, Graz University of Technology

Composition du Jury :

Président :	Katia JAFFRES-RUNSER	Professeur, Toulouse INP ENSEEIHT
Examinateurs :	Mikael GIDLUND	Professeur, Mid Sweden University
	Thomas WATTEYNE	Directeur de recherche, INRIA équipe AIO Paris
	Carlo Alberto BOANO	Professeur, Graz University of Technology
Dir. de thèse :	Nicolas MONTAVONT	Professeur, IMT Atlantique
Co-encadrant :	Georgios Z. PAPADOPOULOS	Maître de conférences, IMT Atlantique
	Samuel CREGUT	Expert Système de Gestion Batteries, Groupe Renault

Invité(s)

Patrick BASTARD	Directeur de la recherche, Groupe Renault
Xiaolin LU	Fellow and Director of IoT Lab, Texas Instruments

Abstract

Electric Vehicles (EVs) have been an active subject of research and development over the years. Because the battery is the most critical element of an EV, it is important to always keep the battery within the optimum temperature and voltage limits. For this reason, a Battery Management System (BMS) is necessary to monitor and ensure the correct operation of the battery pack. In a classic approach, the BMS uses a master-slave architecture with a wired daisy chain link between the Central Processing Unit (CPU) and the sensors. Lately, the industry has seen interest in replacing the wired bus with a wireless link between the master and the slaves. Wireless BMS (WBMS) can bring multiple advantages like easier battery pack fabrication, fewer connection risks, and simpler battery second life management. However, introducing a wireless link in a BMS is a challenging task because we must control the energy consumption of the radio, combat external interference, and avoid the multipath fading phenomenon while keeping the same robustness to guarantee the EV required safety level.

In this thesis, we present a complete solution to enable robust wireless communications inside the battery pack of EVs. We propose an enhanced version of the IEEE Std 802.15.4 Time Slotted Channel Hopping (TSCH) Medium Access Control (MAC) mode running over the physical layer of Bluetooth Low Energy (BLE). To coordinate the data plane, we introduce a reliable and predictable scheduling function based on the Group Acknowledgement (GACK) method to dynamically manage the retransmissions. In addition, we design a special channel blacklisting algorithm to keep the robustness even under high external interference. For the parking state of the vehicle, we propose a WBMS sleep mode that reconfigures the network architecture allowing the central node to switch off, preventing the discharge of the small 12 V battery of the vehicle without damaging the slaves' synchronization. We also discuss the new opportunities that the inclusion of a computing capacity in the slaves brings to BMS software development. We demonstrate that these low-power RF microcontrollers can reduce the central CPU's workload and monitor the cells' estate autonomously. The propositions of this thesis are implemented and validated in real hardware for the battery pack of a Renault Zoe. The laboratory and vehicle test results show that our system provides high reliability, bounded latency, and low power consumption.

Résumé

Les véhicules électriques (VE) ont été un sujet actif de recherche et de développement au fil des ans. Étant donné que la batterie est l'élément le plus critique d'un véhicule électrique, il est important de toujours assurer les conditions optimales de température et de tension. Pour cette raison, un système de gestion de la batterie (BMS) est nécessaire pour surveiller et garantir son bon fonctionnement. Dans une approche classique, le BMS utilise une architecture maître-esclave avec une liaison filaire en série entre les capteurs et l'unité de calcul. Dernièrement, l'industrie a vu l'intérêt de remplacer ce bus filaire par une liaison sans fil. Le BMS sans fil (WBMS) peut apporter de nombreux avantages, tels qu'une fabrication plus facile des batteries, moins de risques de connexion et une gestion plus simple de la seconde vie des batteries. Cependant, remplacer un bus filaire par une liaison sans fil est une tâche ardue, car il faut maîtriser la consommation énergétique de la radio, lutter contre les interférences externes, et éviter le phénomène d'entrelacement de chemins, tout en assurant le niveau de sécurité exigé pour un VE.

Dans cette thèse, nous présentons une solution complète pour rendre possible la communication sans fil robuste dans le BMS. Nous proposons une version améliorée du protocole d'accès au milieu « Time Slotted Channel Hopping » (TSCH) définie par le standard IEEE 802.15.4 en conjonction avec la couche physique de Bluetooth Low Energy (BLE). Pour coordonner l'échange de données, nous introduisons une fonction d'ordonnancement fiable et prévisible basée sur la méthode d'accusé de réception de groupe (GACK) qui gère dynamiquement les retransmissions. De plus, nous concevons un algorithme spécial de liste noire des canaux radio pour conserver la robustesse même en cas d'interférences externes élevées. Quand le véhicule est stationné, nous proposons un mode endormi pour le WBMS qui reconfigure l'architecture du réseau permettant au nœud central de s'éteindre à fin d'éviter la décharge de la petite batterie 12V du VE. Nous discutons également des nouvelles opportunités que l'inclusion d'une capacité de calcul dans les esclaves apporte au développement logiciel du BMS. Nous démontrons que ces microcontrôleurs RF de faible puissance peuvent réduire la charge de travail du processeur central et surveiller de manière autonome les cellules. Toutes les propositions de cette thèse sont implémentées et validées en matériel réel pour le pack batterie d'une Renault Zoe. Les résultats des tests en laboratoire et sur véhicule montrent que notre système offre une haute fiabilité, une latence spécifique et une faible consommation d'énergie.

Acknowledgement

The current thesis results from three years of exciting work at the Renault development center next to Paris. First, I would like to thank my supervisor, Samuel Cregut, who gave me the opportunity to join him in this wonderful adventure of the WBMS. The outcome of this thesis would not be possible without his curiosity for new technologies and his passion for creating disruptive systems. No words can describe my gratitude for teaching me everything I know about the automotive world. Thank you for giving me the necessary freedom in my research and supporting me whenever I needed help.

I could not have undertaken this journey without my thesis advisor Nicolas Montavont and my thesis co-advisor, Georgios Papadopoulos, at the IMT Atlantique. I found our meetings very insightful as you always helped me to view the challenges from another point of view. I feel incredibly fortunate to have had your guidance during these years.

It is an honor to have Mikael Gidlund, Carlo Boano, Katia Jaffres-Runser and Thomas Watteyne as part of my jury. Many thanks for accepting our proposition. It is fantastic to have Thomas on this jury as he introduced me to the beautiful world of low-power wireless communication during my first internship in France.

I would like to express my special gratitude to the BMS team at Renault for this wonderful time. Especially to my boss Cedric Chantrel who has always been available to help me with all the administrative procedures. The results achieved in this thesis would not have been possible without the people at Texas Instruments. In particular, thanks to Xiaolin Lu and Paul Antunes, who have provided us with all the necessary material to implement our proposal.

To my parents, Luz Ofelia and Marco Antonio, my brother Nicolas and my sister Luz Ángela thank you for not missing a single Sunday video call. Even though we live on two different continents, I always feel you are very close. Having your unconditional support and love is the biggest motivation to face any challenge. Without any doubt, I would not be any close to where I am without you. Special thanks to my girlfriend, Liz, for your endless support. You have always found the right words to encourage and push me to finish this thesis. You have made this adventure a lot more incredible. Big thanks to our little cat "Pancita" who has made the writing process much more fun. Finally, thanks to all my friends who have supported me during this adventure; it would not be the same without you.

Table of Contents

Chapter 1

Introduction

Electric Vehicles (EV) has been an active subject of research and development over the years, as they are considered the present and future of transportation. We may think that the electric transportation history is relatively short. However, the first Direct Current (DC) electric motor dates back to 1832, and electric trolleys were first tested in St Petersburg, Russia, in 1880. The first regular tram was installed in a suburb of Berlin in 1881, the same year the French engineer Gustave Trouvé built the first EV [1][2]. At the beginning of the 20th century, EVs were considered a strong contender for road transportation. In 1900 38 % of the American automobile market was caught by EVs [3]. Compared to the Internal Combustion Engine (ICE) based vehicles, EVs were more reliable, less smelly, and started instantly without needing an external force. Unfortunately, there was no affordable way to store the necessary energy to travel long distances. When cheap oil was available, ICE vehicles were preferred. In the middle of this century, electric trains were developed worldwide as the whole system was remarkably reliable, and they did not have the storage problems of EVs as they used overhead lines to power their motors. On the other hand, EVs took a secondary role and were used only in small applications where pollution and noise would be unacceptable, such as warehouses or golf courses [1].

At the end of the century, the environmental concern about the consequences produced by ICE vehicles gained relevance, and some actions started to appear from public entities. For example, in 1900, the California Air Resources Board (CARB) adopted the Zero Emission Vehicle (ZEV) mandate. The ZEV indicates that the vehicle market in California share must include 2 % EV in 1998, 5 % in 2001, and 10 % in 2003. However, the vehicle manufacturers claimed that it was impossible to offer cheap electric vehicles to meet the requirements, which caused the abandonment of the ZEV [4]. Today, the environmental question is much more important than 30 years ago. In December 2015, the Paris agreement was adopted by the Paris climate conference and ratified by the European Union in October 2016. The agreement is a global framework to limit global warming below 1.5 $^{\circ}C$ and avoid dangerous climate change [5]. One of the goals of this agreement

is to reach greenhouse net-zero by 2050, which means cutting greenhouse gas emissions as close to zero, so it should be possible for the atmosphere to absorb the remaining. In this scenario, electric mobility should be essential in achieving the agreement's goals. In 2022, the EU assembly voted to ban the sale of new non-full EVs from 2035 [6]. Today, France has almost 1 million EV and plug-in hybrid vehicles [1]. According to the "Réseau de Transport d'Électricité" (RTE), which is in charge of the operation of the French high voltage system, it would be 4.8 million in 2028 and up to 16 million in 2035, which would represent nearly 40 % of the total units on the road [7]. Even though an EV does not mean zero emissions, the AVERE affirms that during its lifecycle, it reduces by 77 % CO2 emissions with respect to an ICE vehicle. Even if the EV production releases more CO2 than an ICE vehicle, the RTE affirms that the EV only needs to travel between 30000 and 50000 km to compensate for this difference, which is much lower than the average utilization of a vehicle (200000 km in France). The RTE concludes in their report of 2019 that EVs have a carbon footprint 2 to 4 times lower than ICE [7]. Within this context, EV development efforts have been multiplied worldwide to increase their offer and make them more attractive to the public.

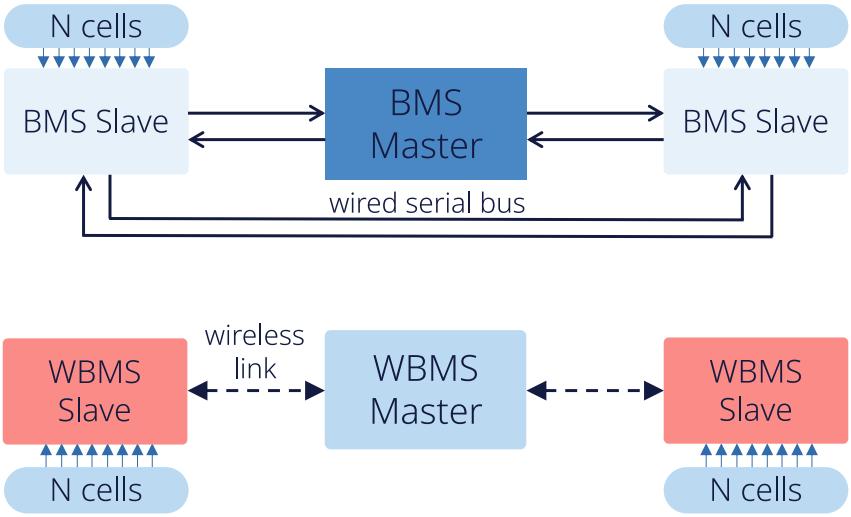


Figure 1.1 – From wired to wireless BMS in EV.

Throughout the history of EVs, the main aspect where they lost the fight with ICE vehicles was autonomy. Undoubtedly, the development of electric batteries capable of

1. www.avere-france.org/

storing the necessary energy to allow EVs to travel an acceptable distance was the trigger to set them at the center of the transportation revolution in the years to come. In the first part of the 20th century, there was not much innovation in the electric batteries field. In the late 1960s, medical and military applications, together with the explosion of the electronic consumer market, demanded portable energy solutions [8]. A significant breakthrough arrived with the utilization of metallic lithium in the batteries, as this material assures a much higher specific capacity than the options available. The first lithium-based rechargeable batteries appeared in late 1970. However, multiple operation failures, like fire incidents, showed that ensuring safe and long operations of these batteries is challenging. In 1991, the Japanese manufacturer Sony introduced a battery with a proper material combination for the electrodes, graphite for the anode, and lithium cobalt oxide for the cathode. This battery demonstrates that using a Li-contained oxide rather than metallic lithium helps to obtain a safer operation [8]. Battery development is an active research subject as it is the most constrained element towards standardizing EVs.

1.1 Wireless battery management system

Today, almost all EVs on the road use Li-based batteries. This kind of receptacle has many advantages, such as its power density. However, they also have dangerous behaviors if their operation conditions (i.e., voltage, temperature) go out of the allowed limits. For this reason, every EV requires a Battery Management System (BMS), a set of sensors, and actuators in charge of monitoring and ensuring the correct operation of the battery pack. A good implementation of this system can avoid catastrophic accidents and extend the battery lifetime. Typically, a battery pack is composed of cells grouped into modules. The BMS periodically measures the battery cells' voltage, temperature, and current to estimate their State of Charge (SOC) and State of Health (SOH). Based on these calculations, the system will allow or disallow the battery charge and discharge, alert the vehicle about an abnormal battery behavior, or set up the balancing process to equalize the charge level among the battery pack. Even though the BMS history is relatively short, its architecture has evolved over the years to increase its performance and optimize its resources. Today, the BMS uses a distributed topology where the processing unit and the sensors are separated in a master-slave architecture. The slaves contain a BMS ASIC to measure the voltage and temperature of the cells, while the master has the CPU, which collects the data and makes the calculations. The system uses a robust daisy chain wired

interface to link the master and the slaves to ensure a good performance in such a critical system.

Lately, the industry has seen interest in removing the wired bus and implementing a wireless link between the BMS master and the slaves. Even if it is not usual to rely on wireless communication in critical applications, the wired version has some significant disadvantages. For example, the copper wiring harness is heavy and occupies space inside a battery pack which would be better to fill with battery cells [9]. It also reduces the battery pack design flexibility and increases its complexity, which results in additional time (money) spent in the development phase. We also need to consider that even if the wired protocol is robust, the system should use connectors between the devices that can suffer from mechanical failure [9]. A typical connector failure is produced by intermittent or high contact resistance. When they are new, the resistance value is minimal ($1-3\ m\Omega$). However, temperature changes (very common in a battery pack) and external vibrations produce micromotion between the contacting surfaces causing its degradation (fretting) and increasing the contact resistance [10]. This process would produce communication failures in the long term. Another disadvantage would be that the complexity of the harness makes it very difficult to fully automate the battery pack assembly process [9]. All the previous reasons have opened the discussion of replacing the wired link with a wireless option. Apart from solving the mentioned problems, Wireless Battery Management System (WBMS) has other significant benefits. For example, diagnosing the battery modules can be done without touching the pack, and replacing the faulty ones would be easier. The WBMS would allow a more straightforward battery second life management, as the wireless slaves can easily change from one network to another, and it would not be necessary to create a new wire harness for every second life application [9]. Finally, introducing a wireless network means that every device should have a microcontroller to manage the data exchange, which could also be used to save the historical status of the cells or even execute some BMS algorithms.

However, replacing a wired bus with a wireless link in such a critical application is challenging. Wireless communication uses a naturally unstable medium and is exposed to external interference. A WBMS network solution must guarantee the EV required safety level, ensuring high reliability, bounded latency, and keeping as low as possible the extra energy consumption with respect to the wired version. Since the WBMS has very special requirements, the network solution must be carefully designed. Among the multiple wireless network options available, none of them was specially designed for this

application. Nevertheless, some of them may be adapted or combined to meet all the requirements. The IEEE Std. 802.15.4 is the referent document for Low-Rate Wireless Personal Area Networks (LR-WPAN), which has evolved during the years, enhancing its performance and robustness to reach several use cases. Another option would be Bluetooth Low Energy (BLE), widely used for multiple applications. These options use time division and frequency diversity mechanisms to achieve low power and highly reliable communication. Wireless IO-Link has been created especially for industrial Wireless Sensor Networks (WSN) needing low latency and high reliability. ZigBee has been developed over the IEEE Std. 802.15.4 and is mainly used for home automation, hence it is very easy to find compatible devices in the market. Thus, there is an open discussion about the most appropriate wireless network for the WBMS. In this thesis, we intend to join the debate presenting all the relevant aspects of the problem, proposing a solution based on state of the art and demonstrating its feasibility with real-world tests.

1.2 Thesis contribution

This manuscript intends to discuss the challenges around WBMS adoption in the automotive industry. It provides some insights to solve the implementation problem while keeping the necessary robustness of such a critical system. We aim to propose, implement and test a WBMS solution to evaluate the convenience of introducing this new system in EV. The contributions contained in this document are as follows:

— **Background and state of the art:** The work of this thesis brings together two very different subjects; EV and low-power wireless networks. We provide a state-of-the-art review on EVs, electric batteries, and BMS. We detail the safety requirements and risks related to bad manipulation of these kinds of systems. We also explain the working principle of the major wireless protocol candidates to implement a WBMS network. Additionally, we present the available options in the market related to the WBMS.

— **Robust wireless communication for critical applications:** We propose a framework that can be adapted to multiple WSN applications requiring high reliability, bounded latency, and low power consumption such as the WBMS. Our proposal is based on the IEEE Std. 802.15.4 TSCH MAC mode with a custom scheduling function that uses the Group Acknowledgement (GACK) method to enable dynamic retransmission management. The proposed network uses the BLE

physical layer and channel blacklisting to guarantee high performance even in environments with high external interference. We also present some hardware and software considerations to take into account when implementing our WSN on different platforms.

— **Autonomous sleep mode for WBMS:** We present a low-power mode for our proposed network, which meets all the requirements of the WBMS. Our approach consists of switching from a centralized star topology in active mode to a distributed architecture in sleep mode. In our sleep mode, the nodes can keep synchronization and the battery pack balanced without any intervention of the WBMS master, which avoids the undesired discharge of the EV 12 V battery.

— **BMS software architecture revolution:** The introduction of a WBMS network means that it is necessary to add a microcontroller to each slave to manage the data exchange. It means that we have additional computing power compared to the wired version, which is an opportunity to modify how we implement BMS software. In this thesis, we propose and test some new features that take advantage of this new distributed architecture and the synchronization given by the wireless network.

— **WBMS vehicle test:** We present a study of the compatibility of the WBMS with the other RF systems of an EV. We also measure the effects of external interference in the link stability of the network when the battery pack is inside the vehicle. Finally, we compare the network performance when the vehicle is in driving and parking mode.

1.3 Thesis context

The work of this thesis has been developed under the framework of a CIFRE convention, which is a French program to promote the development of research partnerships between academia and industry [2]. In this case, the Renault Group, which is an EV manufacturer pioneer in Europe joins forces with the OCIF team at the IRISA (Institute for Research in Computer Science and Random Systems), which has a strong knowledge on low power wireless network, to evaluate and propose a solution for the challenges that the WBMS represent.

Even though the propositions presented in this document can be applied to multiple use cases, we used the Renault Zoe vehicle as the target to implement and prove our

2. www.anrt.asso.fr/fr/le-dispositif-cifre-7844

theoretical solutions. The Renault Zoe battery pack comprises 96 cells grouped into 12 modules. The total capacity of the battery pack is 52 kWh and can reach up to 400 V [3]. The requirements of its wired BMS are:

— The BMS requires a new voltage and temperature measure from all the battery cells every 100 ms.
— The master must send a new balancing command every 1 second.
— The system must switch from active to sleep mode and vice versa in less than 1 second.

1.4 Outline

This thesis is organized into seven chapters. Chapter 2 presents the technical background and state-of-the-art on EV, electric batteries, BMS, low-power wireless networks, and WBMS. Chapter 3 describes our proposal to enable robust wireless communication in critical applications such as the WBMS. In Chapter 4, we propose a WBMS sleep mode where there is no need for the BMS central microcontroller. In Chapter 5, we discuss which new features are possible with the inclusion of the WBMS in EVs thanks to the new computing power added in the slaves. Chapter 6 presents the results of our Renault Zoe WBMS test campaign. Finally, in Chapter 7, we summarize our work, present the conclusions and discuss the perspectives for future work around WBMS.

3. www.renaultgroup.com/

Chapter 2

Technical background and state of the art

The Battery Management System (BMS) is a critical component of an Electric Vehicle (EV) as it ensures the optimal operations conditions of the battery pack. Enabling wireless communication between the sensors of the BMS is a demanding task as it is necessary to implement a highly reliable network with deterministic latency and low power consumption. Before designing and developing our Wireless Battery Management System (WBMS), it is essential to understand the challenges of introducing a wireless network in such a crucial system. To do this, we need to comprehend the associated risks of a battery pack, which options we have in the state-of-the-art of wireless networks to implement a WBMS, and the approach taken in previous works. This chapter presents an overview of EVs, electric batteries, and traditional BMS in section 2.1. Then, section 2.2 presents and compares the available options in the literature to create a low-power wireless network that could be used in a WBMS implementation. Section 2.3 reviews the previous academic and commercial development of WBMS. Finally, section 2.4 summarizes this chapter and presents the conclusions to set the basis of our proposal development described in the following chapters.

2.1 EV and Battery Management System

The battery pack is an essential and potentially dangerous component of electric vehicles. For this reason, the BMS is a crucial system that must be robust in several conditions. In this section, we first introduce the general EV architecture to understand the need to include a battery pack and how it interacts with the other components of the vehicle. We also explain the basic principle of electric batteries and why they represent a safety risk when the optimal operation conditions are not respected. Then, we give a detailed description of a classic BMS, its main functions, the architecture, and principal algo-

rithms. Furthermore, we present the EV safety concept and how this principle must guide the BMS design. We close this section with a discussion of the motivations to develop a WBMS for EV and the challenges the design must consider.

2.1.1 EV architecture

EVs do not have as many complex mechanical arrangements as ICE vehicles do. The motor is the only moving part, connected through electrical wires to a power source, generally a DC battery. Because both pieces do not need to be next to each other, EVs are flexible and can have multiple possible configurations depending on the application [11]. Figure 2.1 presents an overview of the architecture of an EV [12].

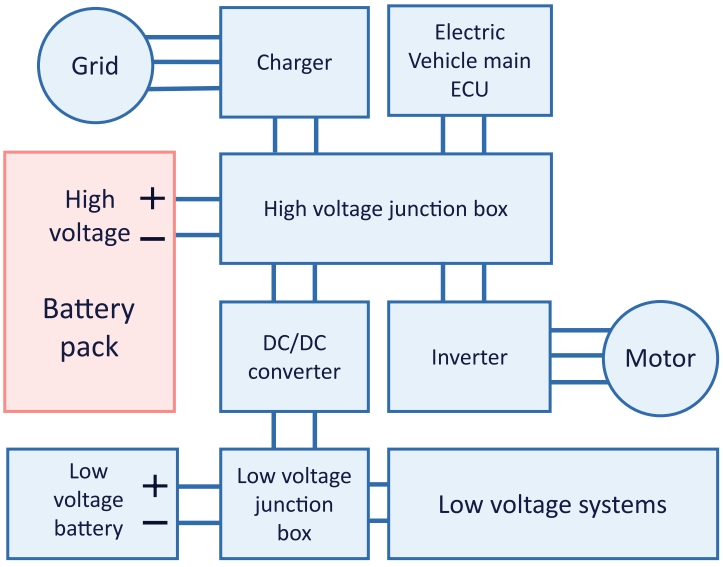


Figure 2.1 – General Electric Vehicle architecture [12].

High Voltage (HV) DC batteries are the easiest way to store energy. The HV battery pack includes a group of cells and a BMS, section 2.1.2 and 2.1.3 provide a detailed description of these components. The motor is in charge of converting electric energy into mechanical energy and vice versa. The main idea is to turn the rotor using a rotating Magnetic Field (MF) generated in the stator by an Alternating Current (AC) [11]. Therefore, a three-phase DC-AC inverter is necessary. Figure 2.2 shows how this device receives DC voltage as input and returns three AC voltage signals phase-shifted by $\frac{2}{3}\pi$, which produces the necessary rotating magnetic field [1][13]. EVs can have multiple configurations. The

Nissan Leaf or the Renault Zoe has one motor mounted at the front. The Tesla Model S has a single motor version mounted in the rear and another with two motors providing all-wheel drive [11].

The charger in the EV receives an AC input from the grid and returns a DC output used to charge the battery. External charging stations have more powerful, bigger, and more expensive AC-DC converters to allow fast charging. As in ICE vehicles, there is also a Low Voltage (LV) battery to power the control unit and other LV systems. Therefore, it is required to include a DC-DC converter to charge the LV battery from the HV system [12]. The junction box is in charge of controlling how the energy flows between the systems. It can allow/disallow the charging process from the grid, the motor alimentation, or the LV battery charge process. The junction box is controlled by the EV main Electronic Control Unit (ECU) based on the information obtained from the BMS and the other systems.

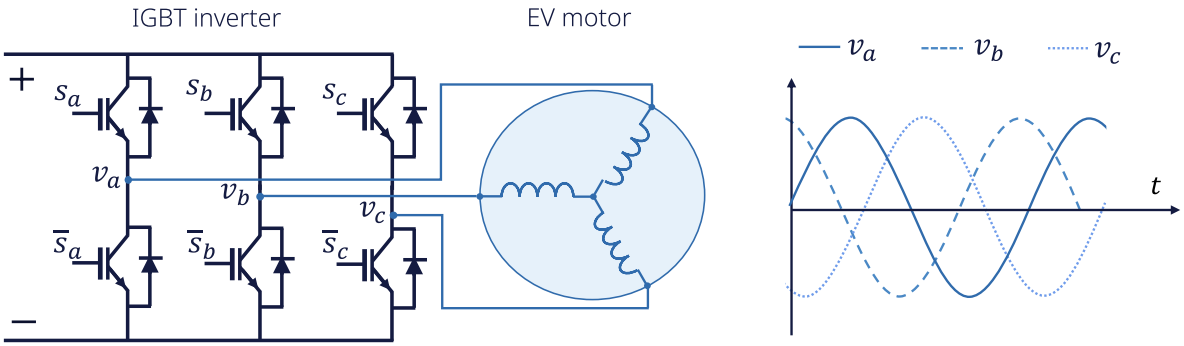


Figure 2.2 – DC/AC inverter [13]

Quick introduction to electric motors

Not all EVs use the same type of motor, but generally, the industry prefers to generate the variable magnetic field in the stator to turn the rotor. A common approach is to install some coils in the stator and power them with altering current to produce a rotating magnetic field. Then, we can use Permanent Magnet (PM) in the rotator to create a constant MF that will rotate according to the direction of the stator MF. With this configuration, the frequency of the AC source can control the rotation speed of the motor. Figure 2.3a shows a simplified representation of this machine [1]. An example of this kind of motor is the Permanent Magnet Synchronous Motor (PMSM) used in many different EV models like the Nissan Leaf or the Kia Soul EV [14]. Nevertheless, using PM in the rotator

is not the only way to achieve rotation in the presence of the stator MF. Induction motors use a cylindrical cage for the rotor composed of electrically linked copper or aluminum rods. The rotating MF induces an electric current in the conductors of the rotor, producing a force on the rods which turns the rotor. Figure 2.3b illustrates an example of this kind of motor [1]. The Tesla Model S is an excellent example of an EV using this motor [14].

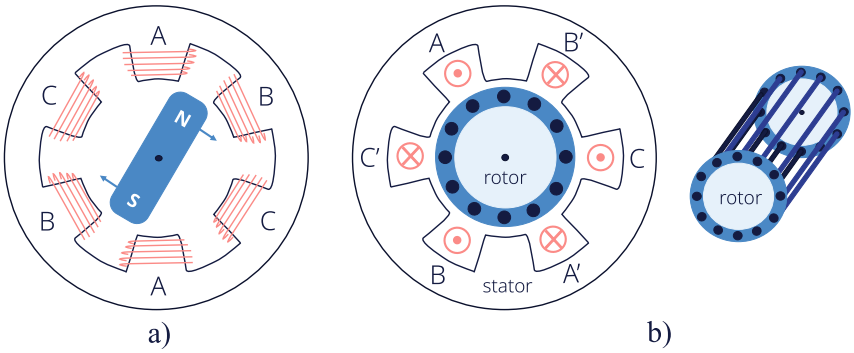


Figure 2.3 – a) Permanent magnet motor, b) Induction motor with cylindrical cage for the rotor [1]

Another option to create an MF in the rotor is to use some coils powered by a DC source. This implementation is called Externally Excited Synchronous Motor (EESM), which the Renault Zoe uses [15]. The main advantage of the EESM is that we can control the current and, therefore, the intensity of the rotator MF. This type of motor has become more popular as it does not need PM, which has very volatile prices [16]. Because of the nature of these three-phase systems, there are some inevitable emissions of Electromagnetic Interference (EMI). Several studies have been done on this topic; for example, Revol et al. show in [17] that the peak EMI emissions of a three-phase AC power train are below 100KHz, but they can go up to 30MHz depending on the type of motor.

2.1.2 Electric batteries

The DC battery pack is a crucial component of EVs, as it stores the necessary energy to power the motor. It is composed of cells connected in series or parallel that transform chemical energy into electrical energy [1]. Alessandro Volta invented the first non-rechargeable cell in 1800. It was a stack of alternating copper and zinc disks separated by washers soaked in brine, allowing the conduction of the current. The first type of rechargeable cell was invented in 1859 by Gaston Planté and is known as the lead-acid battery, which is widely used today in all kinds of vehicles for 12V power [18]. A modern

EV battery pack can reach up to 800V with a weight of more than 200Kg, thus occupying the entire lower part of the vehicle.

Almost all EVs use Lithium-ion-based Battery (LIB) cells with different material configurations. Lithium (Li) is an excellent option for batteries because it is widely available, non-toxic, and is highly prone to lose electrons [19]. LIB cells have the highest energy density, low self-discharge rate, and negligible memory effect [20]. Figure 2.4 illustrates a typical LIB cell that comprises the cathode, the anode, and the electrolyte. The latter prevents the circulation of the electrons within the cell but allows the passage of the Li-ions. Lithium-ions are removed from the cathode and migrate towards the anode through the electrolyte for a charging process using an external source. Meanwhile, the charge balancing electrons must go to the anode through the external circuit. During discharging, the process is the same. When an external load connects both terminals, the electrons use it to go back to the cathode (electric current), and the Li-ions through the electrolyte, as presented in Figure 2.4 [21].

Typically, a Li-contained oxide like the LiCoO2, the LiMn2O4, or the LiFePO4, compose the cathode. The industry uses layered graphite for the anode to facilitate the Li intercalation [21]. The electrolyte combines lithium salt and organic solvents. The second is crucial to increasing the Li-ions mobility and battery performance. A safety separator is in the middle of the electrolyte to prevent an eventual short circuit in the battery. The most common materials for this component are polyethylene and polypropylene, which are permeable to Li-ions [21]. There are multiple standard configurations for LIB cells. An example is the cylindrical setup where the cathode, anode, and separators are rolled and sealed in a metal can. We can also find the stack arrangement, where the three elements are laminated in a film [19].

Despite its significant advantages, lithium is highly reactive, making it challenging to build a safe system with this element. The first measure designers had done was not to use metallic lithium but compounds capable of donating Li-ions, as mentioned before [19]. However, LIB still has many safety issues under abnormal abuse conditions. Mechanical abnormal conditions like crush or penetrations could lead to immediate failure and eventual thermal runaway. In this situation, the battery temperature increases until reaching critical levels causing fire or explosion [21]. Electrochemical abuse like overcharge, overdischarge or short circuit can generate dendritic lithium on the anode, current collector dissolution, exothermic decomposition of electrolytes, thus gas generation, heat generation, and thermal runaway [21]. Finally, batteries may experience external temperature

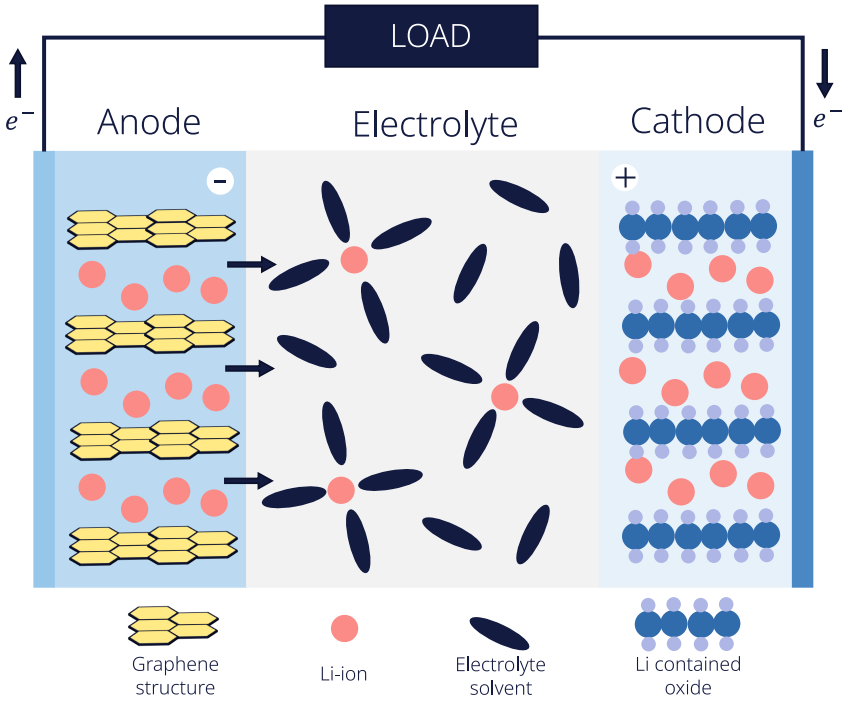


Figure 2.4 – Li-ion based cell representation connected to a load[21].

abuse, rapidly leading to thermal runaway for high temperatures [21] or capacity loss for extreme low temperatures [22]. The safety window concept, illustrated in Figure 2.5, is a two-dimensional space of optimal temperature and voltage values where a LIB cell should stay. Going out of the safety window limits may lead to significant permanent degradation of the battery cell performance or even catastrophic results in the worst case [19]. To build a system, like an EV, that does not represent a risk for the user with a Li-based battery pack, a BMS is mandatory. The BMS monitors the operating condition of the battery and keeps them within the safety window. The following section details the BMS's primary functions, architecture, and principal algorithms.

2.1.3 Battery Management System

In a typical application, the battery pack of an EV is divided into modules that are composed of cells. For example, the Renault Zoe battery pack has 96 LIB cells grouped into 12 modules. As detailed in the previous section, LIB cells represent a safety challenge because abnormal conditions may produce catastrophic consequences. For this reason,

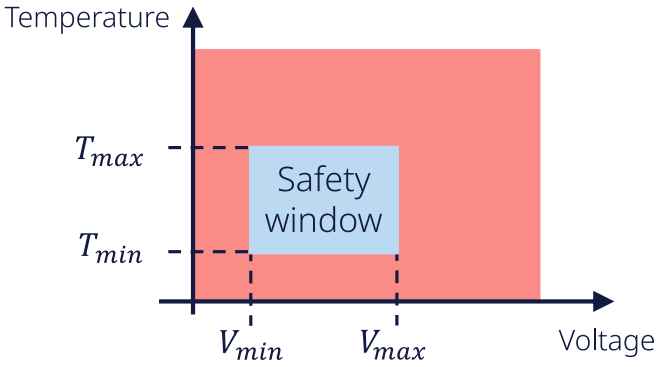


Figure 2.5 – LIB cell safety window [2].

every EV on the road has a BMS, which is in charge of monitoring and controlling the operating conditions of the LIB cells to keep them within the safety window and extend their lifetime [23].

The BMS monitors the battery and ensures its smooth operation. It is a group of voltage, temperature, and current sensors that periodically collect data to estimate the State of Charge (SOC) and the State of Health (SOH) of the cells. Based on these calculations, the BMS can allow or disallow the battery charge and discharge, set a maximum power output and input, alert the exceeding of the safety windows limits, or set up the balancing process to equalize the charge of all the cells [23]. A BMS has two modes of operation, the sleep mode is a low power status used when the vehicle is off, and the active mode is used when driving. Figure 2.6 presents an overview of the BMS algorithm function sequence in active mode. The primary operations are described in the following sections, together with the BMS architecture.

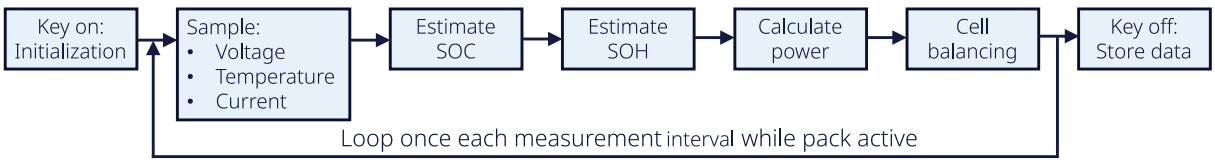


Figure 2.6 – Algorithm sequence performed by a BMS [24].

BMS architecture

The BMS comprises an Electronic Computing Unit (ECU) and multiple sensors. As the battery cells are connected in series, only one current sensor is required to collect the data. This sensor can be shunt or induction type. On the other hand, the BMS uses several Application-Specific Integrated Circuit (ASIC) to measure the voltage as it is necessary to obtain the data for each cell separately. Depending on the ASIC, one of these devices can measure 18 or more cell voltages. We also use the BMS ASICs to derive the resistance value of some thermistors placed in the battery pack. It is not necessary to measure the temperature of each cell, so we use one sensor for a group of cells (e.g., one per module).

Early designs of the BMS presented a centralized architecture where the ECU and all BMS ASICs were embedded in a single board. As Figure 2.7a shows, this design was not practical as many long wires were required to connect all the battery cell terminals to the sensors. To overcome this problem, a master-slave option was implemented in the next generation of the BMS, presented in Figure 2.7b. This architecture divides the BMS into several components, one master and multiple slaves linked through a daisy chain serial connection. The master contains the ECU, and each slave has one BMS ASIC to measure the voltage and temperature of a set of cells. As in the Renault Zoe, a common practice is to use one slave per module, but some other vehicles may use one slave for multiple modules to save costs. The limit for this is defined by the maximum number of cells the ASIC can measure.

State of Charge (SOC)

The capacity of a battery cell is the total electric charge it can supply before it is fully discharged, starting with the cell fully charged [25]. The standard unit is the coulomb which is the amount of charge delivered by an electric current of 1A during 1 second. As this unit is too small, it is preferred to use the Ampere Hour (Ah) unit. A battery of 20 Ah of total capacity can provide 1 A for 20 hours [1]. The State of Charge (SOC) is the percentage of remaining battery capacity relative to the total capacity. The SOC gives information about how many ampere-hours can be drawn from the cell before fully discharged [25]. It is equivalent to the gas level indicator of an ICE vehicle. While many sensors can accurately measure the fuel level. For EV, the SOC must be estimated by an algorithm using the measurements from multiple sensors [24]. An accurate SOC estimation ensures an operation of the cell within the safety window, preventing catastrophic

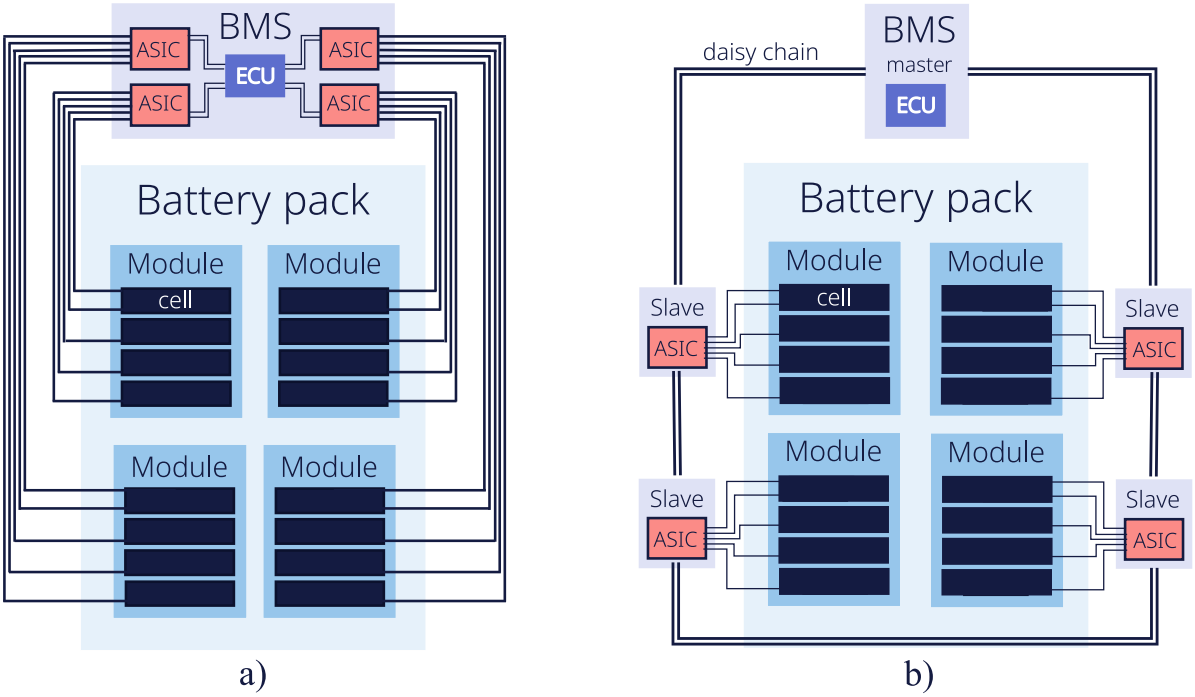


Figure 2.7 – a) BMS centralized architecture. b) master-slaver BMS architecture.

consequences and increasing its longevity [24].

There are two principal ways to compute the SOC of the entire EV battery pack. On the one hand, the BMS can take all the cells as a single unit, average the sensor measurements, and estimate a single SOC value. On the other hand, it can compute the SOC for every cell and report the lowest value when the battery discharges and the highest value when charging, which requires more computing time. For both cases, the most crucial aspect is to use a good estimation algorithm. The following section introduces the Kalman filtering method, widely used in the industry, as it returns very accurate SOC estimation.

A word on Kalman Filtering SOC estimation

There are some options to estimate the SOC of a battery cell. A simple alternative is the measure of the Open Circuit Voltage (OCV) of the cell, but this method only works when the battery is in an unloaded condition, which is impractical for EVs. Another option is the Coulomb-Counting method, which integrates the current drawn from the cell over time. This method requires an accurate current measurement, making it prone to error

accumulation [26][27]. Kalman filtering for SOC estimation is a widely used algorithm as it is a well-known method to estimate the state of dynamic systems [24]. Its main principle is to use a cell model to estimate the SOC and then adjust the computation with the sensor measurements to reduce the error.

In a causal system, the output is a function of the past and present inputs. It is helpful to consider that these systems have a "state" that is not always directly measurable. This state summarizes the effect of all the previous inputs in the system in such a way that it is possible to use it along with the present inputs to compute the present output [24]. Applying this theory to the battery cell, we assume a dynamic system where the state vector includes the SOC, the output is the terminal voltage, and the input is the current and temperature measurements. Note that in this case, the state is not directly measurable. If we can measure the input and the output, the Kalman filtering algorithm allows us to estimate the state of the system [24].

The first necessary element for this method is a system model with two equations. The first one allows us to compute the present state (x_k) as a function of the input (u_k) and the previous state (x_{k-1}). The second one defines the output (y_k) as a function of the input and the state. Figure 2.8 shows an example of a linear discrete-time system model described by equations 2.1 and 2.2 [24]. A LIB cell model depends on the capacity and the materials used for its fabrication. There are many different kinds of models and several methods to define them, such as laboratory characterization or simulation tools. Gregory L. Plett in [25] presents a detailed example of the definition of a LIB cell model.

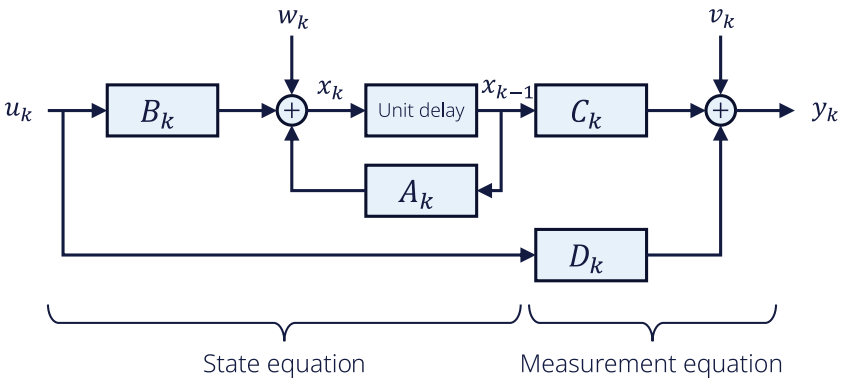


Figure 2.8 – Linear discrete-time system model example. A, B, C and D describe the dynamics of the system and are defined by the model. Sensor and process noise are represented by v and w [24].

$$x_k = A_k x_{k-1} + B_k u_k + w_k \tag{2.1}$$

$$y_k = C_k x_{k-1} + D_k u_k + v_k \tag{2.2}$$

Once the model is defined, we can use the Kalman filtering algorithm to estimate the SOC, as it is presented in Figure 2.9. To do so, at each time the method receives the measurements of current, temperature, and voltage, it computes two different estimate values of the state. The first one uses equation 2.3 of the model with the previous estimated SOC ($\hat{x}_{k-1}$) and the new inputs (u_k) to predict the present state ($\hat{x}_k^*$), then it uses this result to compute an estimated output ($\hat{y}_k$) using equation 2.2. For the next step, we compute the difference between the calculated output ($\hat{y}_k$) and the new received measurement (y_k) to correct the estimated SOC ($\hat{x}_k^*$) and find the new SOC value ($\hat{x}_k$), as it is described in equation 2.4.

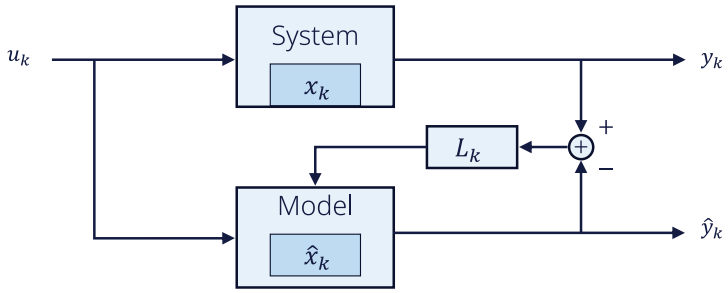


Figure 2.9 – Kalman filtering diagram of state update. The system has a state x_k and the model tries to estimate the state $\hat{x}_k$. [24]

$$\hat{x}_k^* = A_k x_{k-1} + B_k u_k \tag{2.3}$$

$$\hat{x}_k = \hat{x}_k^* + L_k[y_k - \hat{y}_k] \tag{2.4}$$

It is important to mention that the described process only works for linear systems. As most cell models are nonlinear, it is necessary to use a linearization process at every step to compute the estimation. This new algorithm is called Extended Kalman Filtering (EKF) [24].

State of Health (SOH)

EV batteries have some aging processes that gradually modify their performance. These phenomena are consequences of time and use on the battery. The origins of aging degradation may be mechanical or chemical and depend enormously on the battery materials. Several studies show how different conditions affect the battery by splitting the problem into calendar and cycle aging [28].

Calendar aging is the definitive loss of capacity during storage. The degradation of the battery strongly depends on the storage conditions. For example, high temperatures favor secondary reactions like corrosion and lithium loss, which induce capacity fade [28]. Low temperatures limit these consequences but produce loss of material diffusion and battery chemistry modification [28][22]. SOC level during storage is another critical parameter. Studies have shown that elevated SOC produces higher degradation [28]. Cycle aging occurs when the battery completes charge and discharge process. It is a result of multiple factors like utilization mode, temperature, and current solicitation. A battery in use is more likely to experience exothermic effects facilitated by high temperatures, which produce accelerated degradation. A factor commonly used in the literature is the delta SOC, which is the charge variation in a cycle. Some experiments [29][30] have proven that large delta SOC produces high degradation in the cell performance [28]. There are several notions created to evaluate the aging of a battery cell. The most common is the State of Health (SOH), which the BMS calculates periodically. As defined by equation 2.5, it represents the ratio between the nominal capacity at the moment T and the initial capacity [28].

$$SOH(t) = \frac{nominal\ capacity\ at\ t}{initial\ capacity}\ (\%) \tag{2.5}$$

Another helpful notion is the internal resistance (IR) of the cell. It is vital to have a good estimation because it directly impacts the maximum power we can draw/provide from/to the battery [28]. The inconvenience with the IR is that it is impossible to measure directly and is challenging to estimate as it has very small values (mOhm). Some authors propose using a DC current short-pulse to evaluate the IR by measuring the shift in the cell voltage [31]. Another option is to use the EKF method as with the SOC estimation, but here it is necessary to use a cell model including the IR [32]. The Electrochemical Impedance Spectroscopy (EIS) method has shown excellent results as it uses AC injection to estimate the impedance of the cell at different frequencies [33]. IR estimation is an

active research subject as the industry tries to find a low-cost, high-performance method to include in commercial BMS.

Cell balancing

LIB cells are connected in series to achieve the required power levels in a typical pack configuration. A battery pack is said to become unbalanced when the cells do not have the same amount of charge. Slight differences in the cell's dynamics facilitate the apparition of this phenomenon. These differences may appear due to the fabrication processes, cell materials, or environmental conditions. However, keeping all the battery cells with the same charge level is fundamental to using the total capacity of the battery pack [24].

To better understand the problem, let us assume a battery pack composed of two cells connected in series, with a SOC difference of 15 % at some point, as Figure 2.10 shows. Then, we decide to charge the battery, but because the cells are in series, the charging process must stop when one of the cells reaches its maximum SOC. As you can imagine, the other battery cell will store only 85 % of its full capacity. For the discharge phase, it occurs the same. The EV will be able to use the battery power until one of the cells reaches its lowest SOC. Therefore, we can use only 85 % of the battery pack capacity because of the unbalanced cells, and additionally, we have unused charge stored in the other battery cell, as illustrated in Figure 2.10.

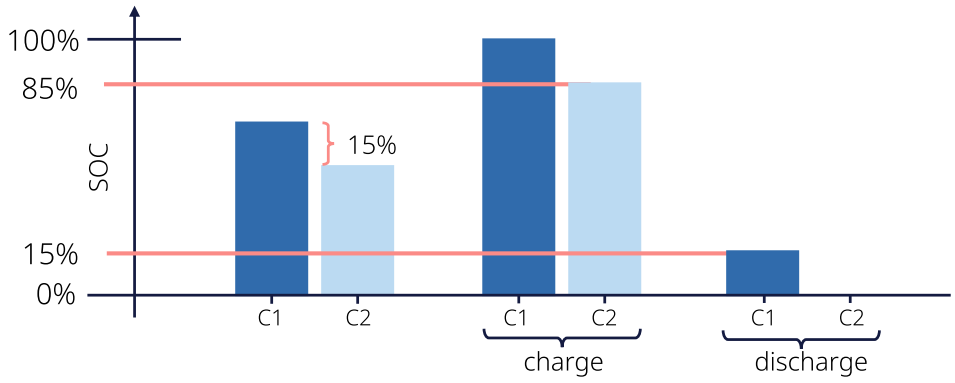


Figure 2.10 – Consequences of having two cells connected in series without the same level of charge.

There are several ways to balance a battery pack [34][35][36]. In the most efficient methods, the BMS transfers the charge of the cells with high SOC to those with low charge. For example, the inductor-based cell balancing technique uses switched inductors

to detect unbalanced cells and transfer the charge [34]. There is also the flyback-based converter technique, which uses a transformer and PWM-controlled switches to move the charge between cells. Unfortunately, these methods require several components, making them expensive for EV applications [34]. For this reason, the industry prefers to use passive balancing, which is less efficient but at the same time less expensive. As Figure 2.11 shows, this approach uses bypass resistors to discharge the cells with high SOC until the battery pack is balanced [34]. The BMS uses transistors in the slaves to enable/disable the cell balancing based on the SOC estimation. In sleep mode, the BMS does not need to compute the SOC and SOH of the battery pack. For long inactivity periods, the primary function of the BMS is to keep the battery pack balanced. To do this, the BMS wakes up periodically (e.g., every 8 hours) to check the charge of the cells and execute the balancing if needed.

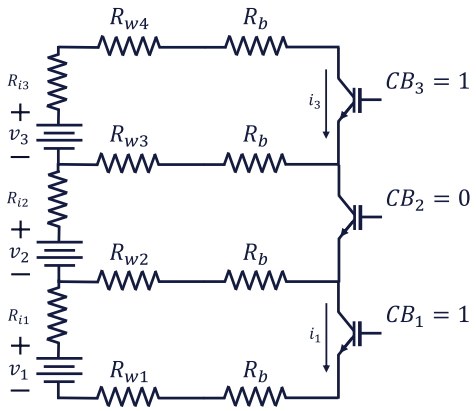


Figure 2.11 – Passive cell balancing circuit example for 3 cells in series. The cell balancing is controlled by the CB signals in the Field-Effect Transistors (FETs). A typical ASIC BMS does not allow consecutive cells to be discharged at the same time.

2.1.4 A word on EV safety

Safety is a crucial objective in automotive development. Functional safety aims to detect dangerous hazardous events that may produce catastrophic consequences to deploy protective measures and mitigations to reduce the associated risks [12]. As the design of functional safety systems is challenging, the standard ISO 26262 was published in 2011 to guide this process. The scope of the standard includes all safety-related electrical or electronic (E/E) systems for automotive applications [37]. It proposes to use a Hazard

Analysis and Risk Assessment (HARA) method where the designers identify hazardous events and specify safety goals to mitigate the hazards [38]. Each safety goal is assigned an Automotive Safety Integrity Level (ASIL) [12]. The ASIL designates the grade of stringency needed to reduce the risk of the item. ASIL A represents the lowest and ASIL D the highest [38].

As mentioned before, a Lithium-based battery represents several risks because improper use can have serious consequences. For this reason, the BMS should be designed as a functional safety system following the ISO 26262 guidelines [12]. According to the EV manufacturer, the BMS must be an ASIL C or ASIL D system. For example, the BMS used in the Renault EVs is an ASIL D system, which guarantees the maximum rigor in the designing phase to reduce the risk associated with the LIB battery. There are several ways to build an ASIL D BMS. Early designs, where the ECU and all the ASIC were embedded in a single board, included two BMS boards to add redundancy and ensure the required safety level. Today, the master-slave architecture uses a different approach. It uses ASIL D compatible ASIC for the slaves, with several diagnostic processes that the BMS can use to ensure its proper functioning. For the master, the BMS includes two ASIL B ECUs. One makes all the calculations, controls the cell balancing, and executes diagnostic routines in the slaves. The other is in charge of monitoring the first to ensure its correct operation.

2.1.5 Towards Wireless BMS

Lithium cells are the most common type of batteries for EV. Keeping a Li-based battery pack within the appropriate temperature and voltage intervals (safety window) is imperative to avoid catastrophic consequences. For this reason, a BMS is mandatory to guarantee the correct operation of the battery pack. A good implementation of this system ensures user safety and increases the battery pack lifetime. A typical BMS architecture uses the master-slave concept where each slave measures the module's data and sends it to the master, which makes the calculations. The slaves communicate with the master using a wired daisy chain. These wires must be isolated with a shield to be immune to the power train's EMI, making them bigger than the reader would expect. Recently, the industry has seen interest in removing and replacing these cables with wireless communication, as it can save space and design time. It would also delete the risks of failures of the physical connections and avoid isolation problems. Introducing a wireless link between the slaves and the master is a challenging task, as this should be highly reliable, deliver the data

within specific intervals, and not spend much energy. The following section presents and compares the available wireless network options we can use to implement such a critical application.

2.2 Low power wireless networks

Even though wireless networks are widely used today, they are not the preferred option for critical systems. The main problem is that the transmission medium is naturally unstable and exposed to external interference. In the state-of-the-art, several wireless protocols are specified to meet special requirements. For example, WiFi networks generally have a high data rate without a particular requirement of low-power consumption or long range. For a WBMS application, it is necessary to implement a network that guarantees high reliability, low-power operation, and bounded latency. Low-power networks are widely used today for multiple applications like home automation (Zigbee, Bluetooth Low Energy - BLE), industrial applications (IEEE Std. 802.15.4, Wireless IO-Link, WirelessHart), or long-range communication (LoRa). This section presents three types of low-power networks that use special methods to enhance the network robustness, making them possible candidates for a WBMS application. The first type is the networks defined by the IEEE Std. 802.15.4, which uses several techniques to achieve high reliability and low power. As it only defines the Physical (PHY) and link layers, these networks can be adapted to multiple applications using different upper layers. The second type is the BLE standard, a complete network stack for low-power communication designed to be adapted to several use cases. BLE shares some concepts with the IEEE Std. 802.15.4, but it uses a different physical layer with higher spectral efficiency. The last protocol presented is the Wireless IO-Link, which is specially designed for industrial applications and relies upon retransmissions to achieve high reliability. Once we have explained the advantages and disadvantages of these protocols, we conclude this section with a comparison of the technologies and the foundations for our WBMS network proposal.

2.2.1 IEEE Std 802.15.4

In 2003 the IEEE Std 802.15.4 was released as the referent document for Low Rate Wireless Personal Area Networks (LR-WPAN). The primary aim was to define a unique physical and data-link layer protocols suite that supports different upper layers mecha-

nisms, to allow mass production of low-cost compliant radio-chips [39]. Over the years, these networks have gained significant ground in industrial applications. However, to meet the specific Quality of Service (QoS) requirements, the standard had to evolve, so in 2012 the IEEE Std 802.15.4e amendment was published. The significant improvement comprises new Medium Access Control (MAC) behaviors to achieve deterministic communication, low energy consumption, and robustness against frequency interference [40].

This section provides an overview of two of the new MAC protocols, Low Latency Deterministic Networks (LLDN) and TSCH. They have several differences, but they use a similar approach to avoid message collision and achieve low energy consumption.

Low Latency Deterministic Networks

The LLDN MAC scheme appears for applications where it is necessary to retrieve sensor data at most every 10ms. It only proposes a star network topology, so a central device exchanges data directly with the peripherals. LLDN relies on the Time Division Multiple Access (TDMA) mechanism to achieve high reliability and low energy consumption. The basic principle of TDMA is to divide the time into small intervals called timeslots. Every node has a specific task for each timeslot; it can transmit some data, receive a message, or go to sleep. The idea is to execute only one message transmission per timeslot to avoid message collisions and increase the probability of successful delivery [41]. TDMA allows the devices to save energy because they can turn off the radio when possible and go to sleep when there is no task.

In LLDN, the standard proposes grouping a set of timeslots into a bigger structure (superframe), which repeats continuously. As you can see in Figure 2.12, there are four types of timeslots. The first is the beacon slot (Bn), where the root sends a message to synchronize the nodes. The second type is the management timeslots, where two dedicated slots are used for the joining process (DownLink and UpLink) [41]. Next, we have the uplink timeslots type, which encompasses dedicated slots for each node to send their data to the root and some shared slots for retransmission of the undelivered messages. The standard indicates that the number of retransmission timeslots can not be higher than half of the transmission ones and that each node can have at most one retransmission timeslot per superframe [41].

LLDN proposes to use the Group Acknowledgement (GACK) mechanism to synchronize the distribution of the retransmission timeslots. With this method, the root does not need to send an individual ACK each time it receives a packet from a node. On the con-

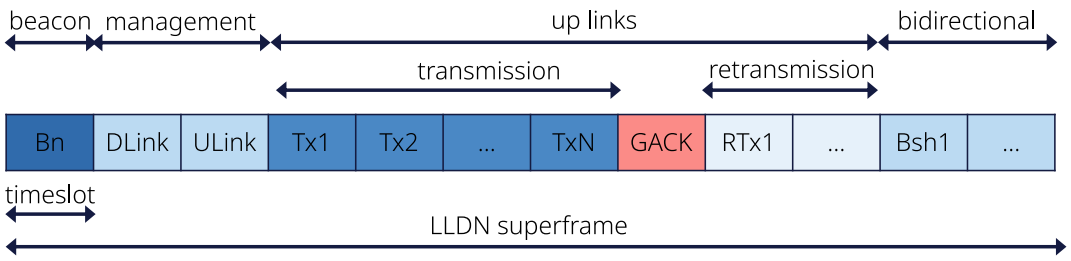


Figure 2.12 – LLDN superframe with dedicated timeslot for GACK [41].

trary, it uses the GACK reserved timeslot to send a broadcast message indicating which packets were correctly received. Based on this GACK, every node knows if they can go to sleep or have to retransmit at a specific timeslot. At the end of the superframe, we have some shared downlink or uplink timeslots that can be assigned by the root at each superframe [41].

Figure 2.13 shows an LLDN example with one root (central device) and five peripherals (A, B, C, D and E). The superframe has five dedicated timeslots for the transmissions, one for the GACK broadcast, and two shared slots for retransmission (RTx1 and RTx2). Assuming that every node sends a new message at each superframe if the root does not receive the message from nodes A and B, RTx1 is given to A and RTx2 is given to B. If the root loses the messages from nodes A, C, and D, the retransmission timeslots will be used by nodes A and C. In this scenario, the message from node D would not be retried. As you can observe, at each superframe the retransmission timeslots are assigned dynamically to achieve high reliability.

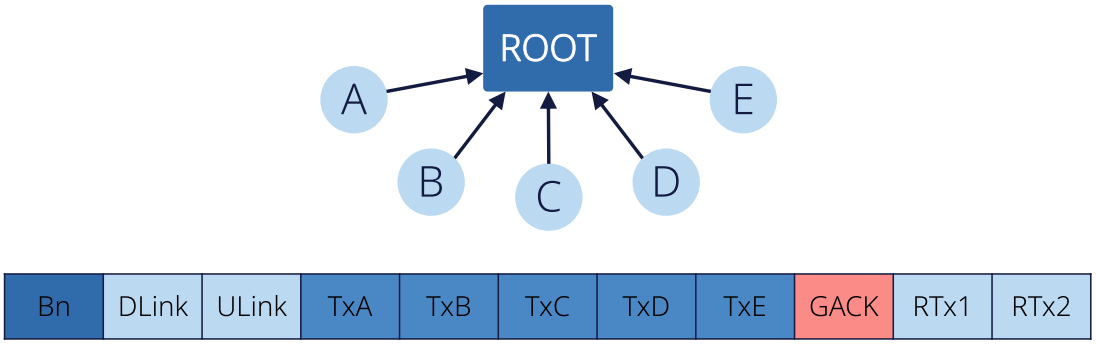


Figure 2.13 – LLDN example with 5 nodes and 2 retransmission timeslots [41].

LLDN uses only one channel for all the message exchanges between the nodes. Thus,

we can have several networks working on different channels in the same physical space. The standard does not propose a timeslot length because it depends on the payload required by the application [41] [42].

Although this MAC mode was removed in the IEEE Std 802.15.4 - 2015 revision [43], there have been several works on these networks to overcome their deficiencies and to adapt them to even more applications. In [41], Anwar et al. present an overview of the proposed LLDN networks in the standards and show some examples about different possible configurations for the superframe. Dariz et al. [44] proposed to use a different length for each timeslot to adapt the network to different payload sizes. They also propose to allocate dedicated links for downlink in addition to the uplink timeslots to reduce latency.

Patti et al. [45], proposes a multi-level multi-channel LLDN network, which uses a tree architecture and data aggregation to have more nodes in the network. Their idea is to execute many transmissions at the same time, over different channels, to maintain low latency. These authors, in [46], propose to dynamically allocate timeslots according to the traffic and the priority of the messages. Their implementation results show that it can increase by 56 % the number of nodes without saturation compared with classic LLDN networks. Using relays to retransmit data on LLDN is proposed by Berger et al. [47]. The principle is that the node sends a message to the root and there is a third node (relay) which is always listening to the channel. If the root does not receive the message, the relay is in charge of retransmitting the frame. Willig et al. [48] use the same approach with relays but they use a reinforcement learning approach to allocate retransmission slots.

The GACK mechanism has been used in several works as it is a good alternative to enhance the performance of wireless networks. Takamori et al. [49] proposes to use GACK on a star topology sensor network. In this work, the network uses a Multi-Code Reception Spread Spectrum Carrier Sense Multiple Access with Collision Avoidance (MCR-SS-CSMA/CA) method, so the master can receive multiple messages at the same time. The problem with this method is that using individual acknowledge would produce a bottleneck in the total throughput. For this reason the authors use the GACK method to inform all the nodes at the same time if their message was received. Chen et al. [50] propose a Group Acknowledgement Strategy (GAS) for internet of vehicles. They use the GACK method to increase the speed of data exchange between vehicles. When a requesting vehicle asks for a file to nearby vehicles, each neighbor responds with a packet, then the requesting vehicle sends a GACK to inform them if the frame was received.

Even though the LLDN MAC mode is no longer part of the latest version of the IEEE Std. 802.15.4, it presents several mechanisms to achieve high reliability and deterministic latency. Especially, the GACK mechanism allows efficient management of retransmissions. Nevertheless, these networks only use a single radio channel and limit the maximum number of retransmissions per node at each slotframe.

Time Slotted Channel Hopping

Time Slotted Channel Hopping (TSCH) is another MAC mode introduced by the IEEE Std 802.15.4 amendment of 2012, supporting star and mesh topologies. This method uses TDMA and Frequency Division Multiple Access (FDMA) to achieve high reliability, and low-power wireless communication [51]. Like LLDN, TSCH splits the time into small intervals, but here, the group of timeslots that periodically repeat is called a slotframe and the timeslot length is set to 10ms. The critical difference between them is that, in TSCH, the network will not use the same channel of frequency for two consecutive timeslots (channel hopping) [51], as you can see in Figure 2.14. Using different channels for each timeslot makes the network more robust to interference and multipath problems, as has been shown by Watteyne et al. in [52] or Muños et al. in [53].

Unlike LLDN, for TSCH, the standard does not propose how the network should schedule the data exchange between the nodes. Instead, it leaves this task to the upper layers. The schedule tells each node what to do (transmit, receive, or sleep) at each timeslot and which channel it has to use [51]. Figure 2.14 shows an example of a TSCH network in mesh topology which has one root and three nodes. The schedule has a slotframe composed of three timeslots, and the nodes can use four different channels. As you can see, in this example, the message exchange from node A to the root always takes place in the first timeslot of the slotframe, but for two consecutive slotframes, this exchange will take place on different channels.

TSCH was included in the IEEE Std 802.15.4 revision released in 2015, and it has been broadly used in different applications [43]. As the standard only defines the physical and link layers, there are some other efforts to standardise a complete stack based on TSCH networks. A good example is 6TiSCH which is the standardisation effort of the Internet Engineering Task Force (IETF) to allow low-power industrial IPv6 networks to run over TSCH [51].

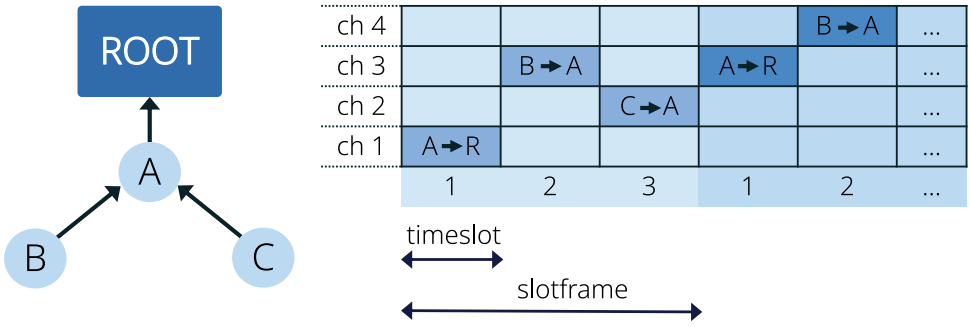


Figure 2.14 – TSCH network example with 3 nodes and 4 available frequency channels.

TSCH schedule

As mentioned, the IEEE Std 802.15.4 does not specify how to build the schedule to manage the exchanges between the nodes. As not all the TSCH networks have the same requirements, using a single algorithm for multiple applications is impossible. Sometimes it is necessary to reduce the end-to-end latency, maximize reliability or minimize energy consumption. For this reason, the TSCH schedule definition is an active research subject, and several scheduling algorithms have been proposed. In this section, we present an overview of some of them. Generally, they are classified into two groups: centralized and distributed. The centralized method has proven more convenient for relatively static networks with deterministic traffic than the distributed approach [54] [55].

Centralized scheduling: In this case, a unique entity (typically the root) creates a schedule based on the network architecture. For example, we can mention the Traffic-Aware Scheduling Algorithm (TASA) [55]. It provides a centralized method to create a schedule based on the bandwidth requested by each node and the network architecture. This algorithm has been created for data gathering networks (WSN) with tree topology [56] [55]. TASA creates the schedule without taking into account retransmission when packet delivery fails. In these cases, the network should wait until the next slotframe to retransmit the message resulting in delays [54]. To add robustness to the TASA algorithm, Gaillard et al. [57] propose allocating additional timeslots for frame retransmission to reduce the packet loss probability. This over-provisioning algorithm uses the link quality (PER) between the nodes to calculate the number of additional necessary resources to achieve a given PDR. The method considers the link quality as time-invariant on the scheduling scale, which does not consider the temporal variations of the link conditions.

Simulation results show that this new proposition increases the reliability of the original TASA algorithm.

The Kausa scheduling algorithm [58] deals with lossy links by scheduling ad-hoc retransmission timeslots. The idea is to assign a route and allocate resources to each client flow using key performance indicators (KPI) such as the end-to-end delay and the PDR. The algorithm first builds a path for each flow, trying to balance the traffic over the network while satisfying the Packet Delivery Ratio (PDR) requirements. Suppose a chosen path does not meet the reliability requirements. In that case, it is blacklisted, and the algorithm recursively looks for a new path until it finds an alternative that respects the delay (success) or all the links are blacklisted (failure). Although Kausa is a good option for developing a schedule that respects delay and reliability constraints, it is not adapted for networks with dynamic conditions [59]. Adaptive Multi-hop Scheduling (AMUS) is another alternative focused on high reliability for WSN [54]. This algorithm assigns redundancy timeslots to weak links as precautionary measures to meet the QoS requirements of critical applications. Simulation results show that AMUS outperforms TASA as it uses more efficiently the spatial diversity accommodating multiple concurrent communications in the same timeslot [54].

Distributed scheduling: For this approach, there are two options; each node can create its schedule autonomously or it can negotiate it with their neighbors. As an example we can mention the Minimal Scheduling Function (MSF), which is the distributed scheduling algorithm adopted by the 6TiSCH stack. In MSF each node monitors the current usage of the assigned timeslots. If the amount of frames that the node needs to send to its neighbor is higher than the capacity provided by the assigned cells, the nodes should negotiate more transmission timeslots for its schedule. On the other hand, if the traffic is lower than the capacity, the neighbors can delete the unused timeslots [51] [60]. This schedule produces some message collisions because the nodes may schedule the same timeslot with the same channel offset. If a node detects a problem with the PDR in a specific timeslot, it renegotiates another link with its neighbor [59].

Accettura et al. [61] present the Decentralized Traffic Aware Scheduling (DeTAS) function for multi-hop low-power networks. This method manages the queues of the devices to avoid overflows while minimizing the network duty cycle. The algorithm assumes a Destination Oriented Directed Acyclic Graph (DODAG) network. Each node informs its parents and the DODAG root of the amount of traffic generated by itself and its chil-

dren. Then, each parent assigns the necessary timeslots to their children. DeTAS achieves low end-to-end latency as reception and transmission timeslots are located consecutively, meaning that the nodes can instantly forward the received packet to avoid buffer overflow. Chang et al. [62] use a similar approach for DAG networks called Low Latency Scheduling Function (LLSF), where the reception and transmission timeslots are located as close as possible to reduce the delay. Nevertheless, the authors consider that the radio links are perfect, so they do not mention how DeTAS or LLSF should manage the retransmission of the lost packets [59].

Hosni et al. [63] propose a distributed algorithm based on stratums that is robust to any change (topology, traffic, routing paths). The idea is to group the nodes according to their distance from the DODAG root. For each stratum, there is a block, a continuous region in the schedule where the nodes send their data to their parents. Successive blocks in the schedule are associated with consecutive stratum. The size of the blocks is not the same as the nodes with the lowest level must forward more data. The stratum algorithm allocates additional timeslots for retransmission according to the link reliability, so this function can deliver the messages within one slotframe. Soua et al. propose Wave [64], a distributed scheduling algorithm for IEEE802.15.4-based networks. The idea is to minimize the latency of the data convergecast by creating a schedule with the shorter slotframe possible. The first Wave is triggered by the sink where the nodes allocate one timeslot by exchanging some control messages with their conflicting nodes [65]. In the second Wave, the nodes are able to set their transmitting and receiving timeslots based on the previous waves. In case of failures, the latency of the packets increases as the schedule does not consider the necessary timeslots for retransmission [59].

The Distributed Scheduling for Convergecast in Multichannel WSN (DiSCA) has also been proposed by Sua et al. [66]. This schedule assigns to each node a set of timeslot and radio channels for each of its transmissions toward the sink. The main objective is to minimize the number of slots and ensure successful transmission within one slotframe. The authors affirm that this method is more performant than its previous version, Wave, as it can take advantage of a sink with multiple radio interfaces. Their document does not discuss frame retransmission or any other packet loss management mechanism. Orchestra is a good example of a scheduling function where the nodes create their schedule without exchanging specific messages with their neighbors (autonomously). It was proposed by Duquennoy et al. [67] for dynamic networks where there is no predefined traffic and nodes can be added or removed. The schedule is created from the RPL neighbor information.

Each node derives the time and channel offsets with a function using the sender's or receiver's identifier. Another example of distributed autonomous schedule is the DIS_TSCH algorithm proposed by Hwang et al. [68]. In this method each node knows their position in the tree (depth of the node and rank among the children). With this information the nodes can allocate their timeslot to send data toward the sink [59].

TSCH network synchronization

The IEEE Std. 802.15.4 presents a timeslot structure where it is possible for the nodes to exchange a message and its acknowledgment frame, as you can see in Figure 2.15 [69]. Because TSCH networks use time division to avoid message collisions, the nodes must be able to exchange data at a specific time. When a node needs to send a message to another, it waits exactly `TsTxOffet` after the beginning of the timeslot to start transmitting the frame, as it is depicted in Figure 2.15 [69]. In an ideal scenario, the receiver turns on its radio and listens to the channel at the same time. In practice, it is impossible for both nodes to execute these operations simultaneously.

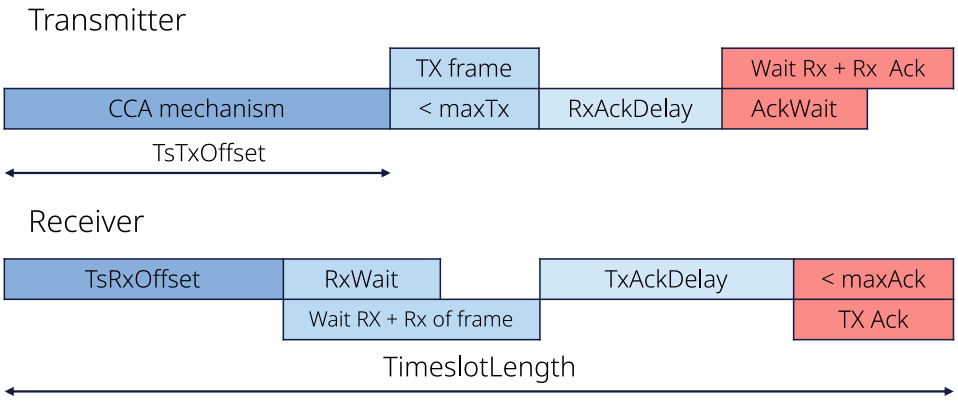


Figure 2.15 – IEEE Std. 802.15.4 TSCH timeslot structure [69].

Wireless nodes use clocks (e.g crystals) as time source. Even if they are the same reference, two different clocks do not have the exact same frequency, which produces a clock drift between nodes [70]. This problem is due to multiple causes, like the manufacturing process, temperature or voltage supply. Typically, manufacturers guarantee a maximum clock drift value expressed in parts per million (ppm) [70]. To overcome this issue, in

TSCH the receiver turns on its radio a little bit earlier, `TsRxOffset` from the beginning of the timeslot, and listens to the radio channel during `RxWait`, which is also called Guard Time [71] [69]. The network must ensure to re-synchronize the nodes periodically to keep their clocks drift below the Guard Time value [71].

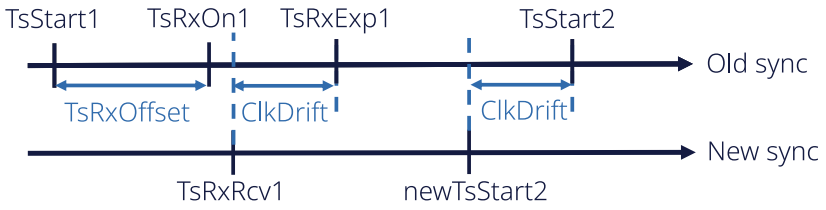


Figure 2.16 – IEEE802.15.4 TSCH receiver synchronization example.

Figure 2.16 shows an example of the synchronization process between two nodes where the receiver adjusts its clock by using an incoming frame from the other node. For the receiver the current timeslot starts at `TsStart1`. It turns on its radio at `TsRxOn1`, and is expecting to receive a frame at `TsRxExp1`. Because both nodes have a drift between the clocks, the message arrives at `TsRxRcv1` which is `ClkDrift` before `TsRxExp1`. With this information the receiver knows that the next timeslot is not starting at `TsStart2` but `ClkDrift` before at `newTsStart2`. Typically, TSCH networks exchange periodically EB messages to resynchronize the nodes, although this process can be done with any kind of frame or with the acknowledgement [71]. The maximum period (Tmax) to re-synchronize the networks is defined by the maximum clock drift reported by the manufacturer and the `RxWait` time defined by the MAC protocol [71]. It can be calculated using Eq. 2.6 where `ClkDriftSec` is the clock drift generated in one second.

$$Tmax(s) = \frac{RxWait \quad (us)}{ClkDriftSec \quad (us)} \tag{2.6}$$

For some applications this `Tmax` could be too short and the synchronization process may consume a lot of energy. The authors in [72] and [73] proposed adaptive synchronization for IEEE Std. 802.15.4 networks. The idea is that the nodes calculate how much their clock drifts per timeslot with respect to their neighbor. With this information they can know how many timeslots are necessary to have a drift of one tick of the clock (30us for a 32Khz crystal clock) [73] . Then, the node sets a counter to this value, reduces it at each timeslot, and, when the counter reaches zero, the nodes apply a one-tick correction to its clock [73] . With this method the nodes can keep synchronization without exchanging

packets every `Tmax` seconds. [70] presents how it is possible to use the adaptive synchronization method to multi-hop networks. In these kinds of networks, each node has a time source neighbor; the authors recommend using the same DAG architecture provided by the RPL protocol, in this way it is possible to avoid synchronization loops. They also remark that when designing a network's synchronization mechanism, it is important to consider the "synchronization swing" phenomenon. To understand this problem, the authors present the example depicted in Figure 2.17. Imagine that we have a network with a Guard Time of 1 ms where node A is the time source neighbor of node B, which is the source of C, and this one is the reference for node D. Suppose that at some point, the clock drift between each node with its time source is 400 us. If we synchronize node B to node A, the clock drift between B and C would be 800 us, which is fine because it is still under 1 ms (Guard Time). Next, if we synchronize node C to B, the drift between C and D would be 1200 us, which would cause a synchronization loss of D as the drift is bigger than the Guard Time [70].

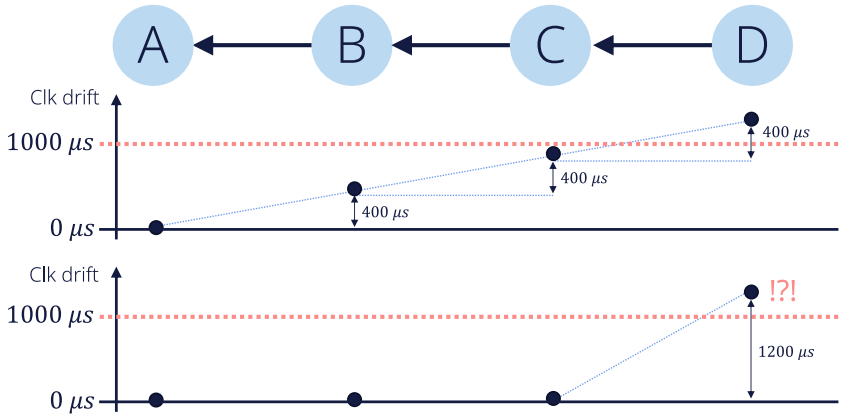


Figure 2.17 – Synchronization swing example [70]. This phenomenon occurs when a node synchronizes with respect to its time source and causes the clock drift with its child to increase over the Guard Time limit, which would produce the child node to lose synchronization.

Papadopoulos et al. affirm that the idle time a TSCH node spends listening to the radio channel during the Guard Time is the most significant cause of energy consumption [71] [74]. Using a mathematical model, they identify the guard time that maximizes the energy efficiency in a single hop TSCH network. They present results from some tests executed in the Contiki OS simulator (Cooja), concluding that fine-tuning the guard time produces significant energy savings without compromising the network reliability [75]. The authors also present in [71] a decentralized adaptation of the guard time to reduce

the energy consumption in a multi-hop TSCH network using an empirical approach. The results have shown that, in a DAG network, the nodes near the sink need a smaller Guard Time value than the furthest nodes. They present an example of a nine-hop network, where the nodes in the first rank use a Guard Time of 650 μs while the nodes in rank 9 need 1200 μs. The authors conclude that the energy consumption could be reduced by 12 % by fine-tuning the Guard Time in a multi-hop network [71].

Temperature is another important variable affecting the TSCH network synchronization as the output frequency of clock sources, like crystals or oscillators, strongly depends on this variable. Xu et al. present TACO [76], a temperature-aware compensation mechanism to keep synchronization under volatile environmental temperatures. The authors model the clock skew over the temperature by solving a constrained least squares problem [76]. Based on this model and the measurements provided by a built-in temperature sensor, the nodes can continuously compute their clock offset and keep synchronization. Martinez et al. also use a model to predict the clock drift according to the temperature [77]. In this work, they propose an algorithm that continuously learns from the measurements and refines the model to increase the accuracy of the estimation. With this approach, the nodes manage to reduce their clock drift to 0.1 ppm, which would reduce the overall energy consumption of the network as the re-synchronization interval can be extended.

Channel blacklisting

Wireless networks' stability extremely depends on the status of the medium. Even if a network uses channel hopping to avoid frequency interference or a good schedule function to enable retransmissions, the network performance is directly affected when the radio channels present high traffic. A common technique in wireless networks is radio channel blacklisting to overcome this issue. The main idea is not to use radio channels that present high interference; in this way, the network would reduce the number of packets lost and retransmissions, save energy and increase end-to-end reliability. A typical blacklisting algorithm has three parts. The first consists of identifying which radio channels have high electromagnetic interference and creating the channel blacklist. A common approach is to use the PDR, described by equation 2.7, rather than the Received Signal Strength Indication (RSSI) because the last one can only be computed for successfully received packets [78]. The second part includes a method to distribute the list to all the nodes

concerned and how to start using it simultaneously. The last part consists in reevaluating the blocked channels to check if the network can use them again. In the literature, there are two principal types of blacklisting methods: global and local. In a global approach, all the nodes in the network use the same channel blacklist. Some authors affirm that these methods are not optimal as the radio links may present different performances according to their locations. In a local approach, each pair of neighbor nodes establish a different list. Although it would be more optimal, it introduces high complexity and may produce message collision within the network [79].

$$PDR = \frac{Delivered\ packet}{Total\ number\ of packets} * 100\ \% \tag{2.7}$$

Kotsiu et al. present LABeL, a link-label adaptive blacklist channel technique for 6TiSCH networks [80]. They use the Window Mean Exponential Weighted Moving Average (WMEWMA) to estimate the link quality as it has demonstrated good performance. The authors affirm that packet losses present a stochastic variable that needs to be smoothed. To do this, the WMEWMA considers the historical state of the link to calculate the present stability. In their proposal, the nodes use the last 16 messages and compute how many of them were correctly received (PDR_{last16}). Then, the algorithm uses equation 2.8 to calculate the present state of the link, using its previous state and a weight factor (α) [80]. This factor determines the dynamic of the estimation, if it is small, the link status will take less into account the historical state of the link, and the estimation presents more fluctuations [81]. Once their algorithm has computed the status of the channels, it uses the one with the highest PDR as a reference to adapt the acceptable threshold and generate the blacklist. The threshold to separate the bad channels from the good ones is adjusted until they obtain at least three acceptable radio frequencies. Once the blacklist is delivered to the nodes, the method to not use the bad channels is simple, if the channel hopping sequence indicates a bad channel for a given timeslot, the node must use the closest good channel of the sequence as a replacement. To recover bad channels, they propose to use sometimes (with a probability p) the bad channels to check if the status is good enough to include them again in the sequence. Their experimental results show that using this adaptive approach improves by 20 % the reliability of a TSCH network.

$$PDR_{wmewma}[ch] = \alpha * PDR_{wmewma}[ch] + (1 - \alpha)PDR_{last16} \tag{2.8}$$

Zorbas et al. propose a channel blacklist method based on the PDR [79]. In this case, they use a predefined fixed threshold to separate good channels from bad ones. They propose creating a new channel hopping sequence using only the good channels each time the blacklist is updated. In this way, the channel selection will always return a good channel directly. In order to avoid channels sequences mismatch within the network, the authors propose to deliver the blacklist with a target Absolute Slot Number (ASN) number (ASN_{BL}) from which the new sequence should be used. For the channel recovery, they propose adding the channels again after some minutes; if there is still interference, the algorithm will put them again on the blacklist. Gomes et al. proposed a distributed blacklisting optimized for multi-hop networks [82]. The authors also prefer using a software-based estimation rather than hardware variables such as the RSSI or Signal to Noise Ratio (SNR). Their proposal uses a channel estimation algorithm based on the Multi-armed Bandit (MAB) problem with an ϵ-greedy strategy. The channel's status is updated using the exponential moving average, where the recent values have more importance than the previous state. Tavakoli et al. propose the Non-Intrusive Channel Quality Estimation (NICE) [83]. Their idea is to select the less noisy channel for the network operation. To evaluate the channel quality, they use silent periods where there is no transmission in the network, and the nodes can listen to the channels to measure the noise. Then, the manager selects the best channels and shares this information using the beacons. The authors also use the WMEWMA to update the quality values with a high weight factor (0.9). Their results from some real experiments show that their proposal provides 24 % higher packet reception ratio than a default TSCH protocol.

Physical layer

The IEEE Std. 802.15.4 proposes multiple modulation techniques for several frequency bands. It is up to the network designer to choose the most appropriate for the application according to the requirements (data rate, range, and bandwidth). There is a tradeoff between range and throughput. For long-range applications, it is preferred to use low-frequency channels (100MHz - 900MHz), which leads to a low data rate. For short-range applications (like the Wireless BMS), the standard proposes to use the 2.4GHz Industrial, Scientific and Medical (ISM) band. The document specifies 16 channels from 2405MHz to 2480MHz with steps of 5MHz between consecutive channels [69]. Figure 2.18 presents a comparison of the channel division in the ISM band for IEEE802.15.4, WiFi, and BLE networks. Note that these three technologies share the same band and may cause interference

if they are in the same physical space.

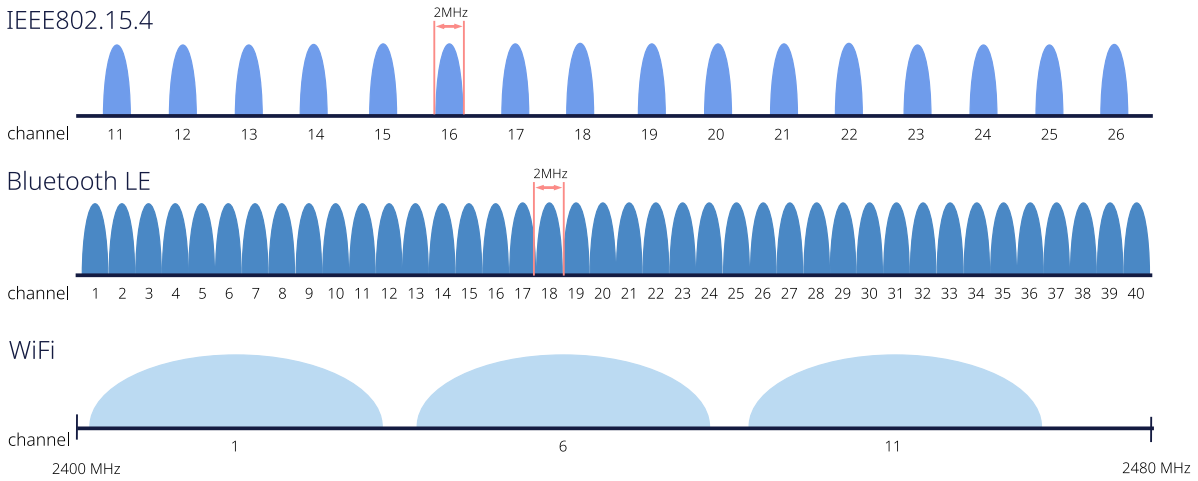


Figure 2.18 – Comparison of channel division in the ISM 2.4GHz band for IEEE802.15.4, WiFi, and BLE networks.

For the 2.4GHz ISM band, the standard uses Offset Quadrature Phase-Shift Keying (OQPSK) [69]. A traditional Quadrature Phase-Shift Keying (QPSK) modulation technique uses a symbol (two data bits) to select the output signal phase among four values. The problem with this method is that it may create 180 degrees phase shifts, producing amplitude fluctuations when it is low pass filtered, which is not convenient when the system uses non-linear components [84]. OQPSK solves this problem by only allowing one bit of the symbol to change each time, as presented in Figure 2.19. In this way the maximum allowed phase shift is $\frac{\pi}{2}$.

Over the signal modulation, the standard also uses a Direct Sequence Spread Spectrum (DSSS) method to make the transmission more resistant to interference. The idea is to use 4 data bits (one symbol) to choose one among 16 pseudo-random noise (PN) sequences of 32 chips, as it is presented in Figure 2.20. Then, the chip sequence is sent over the air using the OQPSK modulation. The 16 PN sequences were calculated to allow the system to detect and correct some false bits produced in the wireless transmission [39]. The chip rate used in this band is $2MChips/seg$, which leads to a maximum data rate of $250Kb/s$ [69].

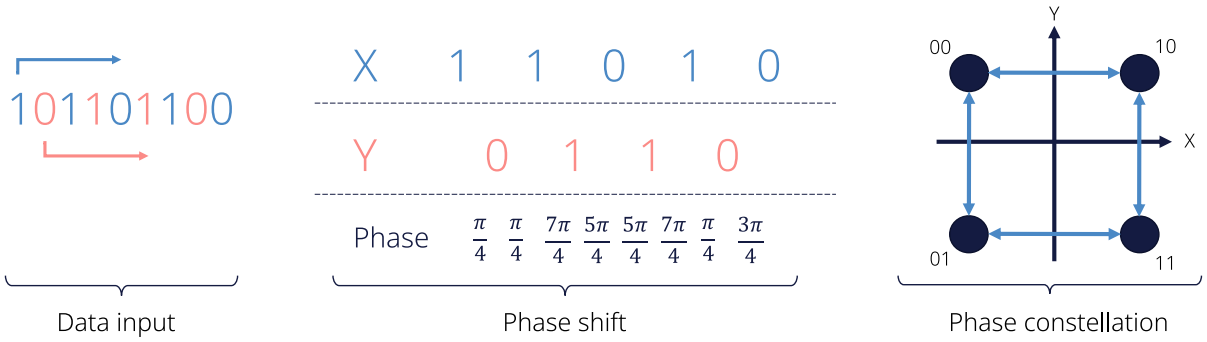


Figure 2.19 – OQPSK chooses a phase value, for the output signal, from the phase constellation according to the digital data input. It allows a maximum shift phase of $\frac{\pi}{2}$ to avoid amplitude fluctuations.

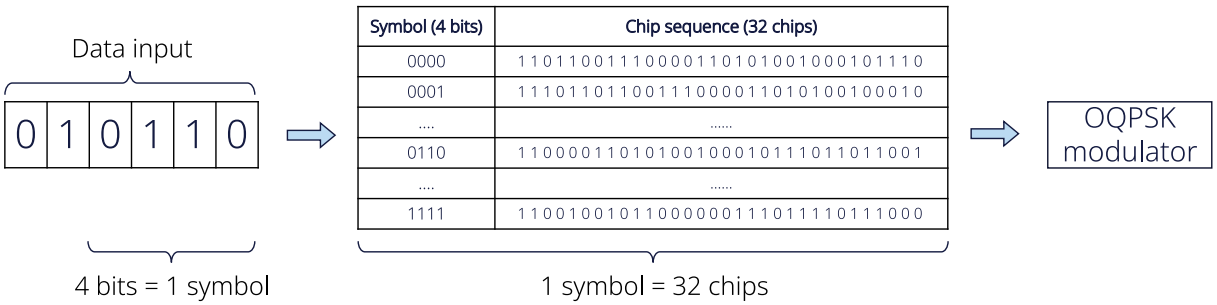


Figure 2.20 – Direct Sequence Spread Spectrum (DSSS) method proposed in the IEEE Std. 802.15.4. In this method 4 bits are represented by 32 chips which allow the system to detect and correct some false bits produced in the wireless exchange.

2.2.2 Bluetooth Low Energy (BLE)

Bluetooth wireless technology was created for short-range wireless communications. It is divided into two types; Bluetooth Basic Rate (BBR) and Low Energy (BLE). The latter targets applications with lower complexity, lower cost, and less current consumption than BBR. BLE operates in the ISM band at 2.4 GHz. It uses TDMA to avoid packet collisions, and frequency hopping to combat external interference [85]. In BLE, the physical channel is divided into subunits called events where communication between devices takes place. For each event the network uses a different radio channel following a specific hopping sequence. There are four types of events: advertising, extended advertising, periodic advertising, and connection events [85].

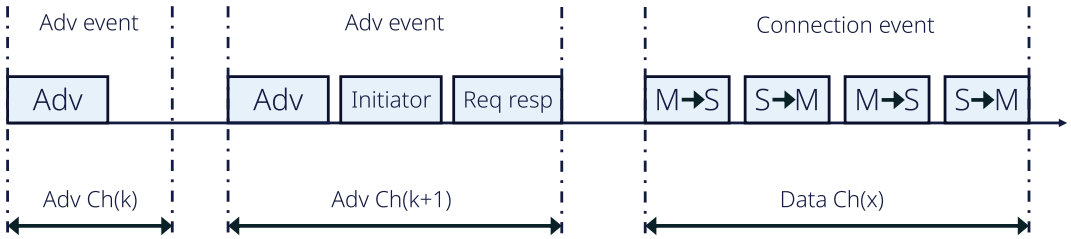


Figure 2.21 – BLE piconet formation with a master (initiator) and a slave.

As depicted in Figure 2.21, we use advertising events to establish a connection between two devices, where an Advertiser periodically broadcasts advertising (adv) packets. At the same time, an Initiator listens to the channel waiting for an adv packet. Once it is received, the device sends a connection request to the advertiser. If the request is accepted, a piconet is created, the initiator becomes the master of the network, and the advertiser the slave device. Then, both devices use connection events to exchange data according to a schedule defined during the joining phase. In a connection event, it is always the master who sends a packet first. If the sender does not receive an acknowledgment from the receiver, it has to retransmit in the following connection events until success. A piconet can be composed of multiple devices with one master at the center which keeps the network synchronization using a star topology. A device can have both roles: the slave in a piconet and the master in another [85].

The BLE specification proposes a complete wireless network stack. It defines multiple entities for each layer as physical channel, physical link, logical transport, logical link and L2CAP channel. These concepts are explained in the specification [85]. This complete set

of protocols allows to set up multiple BLE network architectures. For example, we can create a simple star network with one piconet, a multi-hop network or a mesh network using multiple piconets.

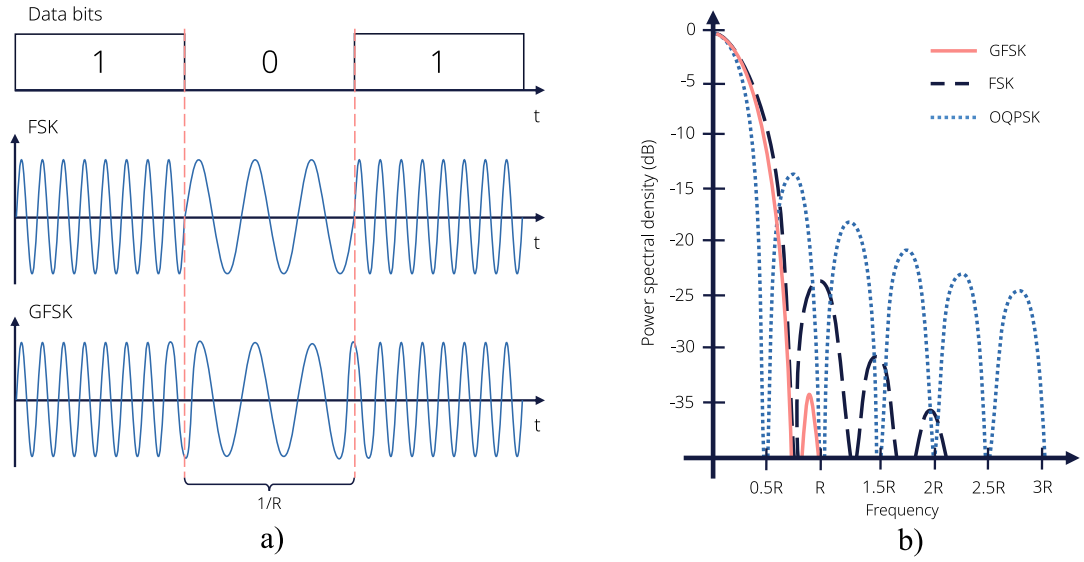


Figure 2.22 – a) Comparison between FSK and GFSK modulation where R is the data bit rate. As you can observe, the frequency shift is less abrupt in GSFK. b) Spectral representation comparison between GFSK, FSK and OQPSK. The figure show that GFSK has the lowest side lobes [86].

Physical layer

BLE uses Gaussian Frequency Shift Keying (GFSK) modulation and operates in the 2.4 GHz ISM band. The mandatory data rate is 1 Mb/s, but the specification proposes coded alternatives to support error correction; $s = 2$ allows 500 kb/s and $s = 8$ for 125 kb/s. There is also an uncoded version with 2Mb/s of data rate [85]. In Frequency Shift Keying (FSK), the modulator changes the carrier frequency according to the data bit. A binary one produces a positive frequency deviation, and a binary zero a negative variation. This modulation is beneficial because it does not require a complex implementation. However, the output signal is not continuous, producing side lobs with high magnitude. Gaussian FSK modulation filters the square input signal to produce a continuous output, which reduces the side-lobes magnitude [86]. This choice allows the specification to use 40 frequency channels separated by 2 MHz between 2402 MHz and 2480 MHz [85]. Figure 2.22a shows a comparison between the output signals of FSK and GFSK. Figure 2.22b

compares the spectral representation of a GFSK, FSK, and OQPSK modulated signal. As you can observe, GFSK has the lowest side lobes among the three options [86].

2.2.3 Wireless IO Link

IO-Link is a standardized IO technology (IEC 61131-9) for wired communication with sensors and actuators. It is defined by the Profibus Nutzerorganisation e.V. (PNO), located in Germany [1] [87]. Typically, industrial automation does not use wireless communication as it has strict low latency and high-reliability requirements, especially for closed-loop control applications. However, wireless links between sensors and control units reduce the installation time and the maintenance operations. To fill this gap, the PNO proposes a wireless IO-Link version, which offers reliable, real-time deterministic communication for factory automation. IO-Link wireless (IOLW) proposes a network with star architecture, so the central device (master) has a direct link with the peripherals (devices) [87]. The physical layer of IOLW is based on the proven Bluetooth 4.2 technology, which works in the 2.4GHz ISM band. It uses GFSK modulation with a data rate of 1Mb/s and 80 channels separated by 1MHz. As you can observe, the IOLW devices can use transceivers designed for Bluetooth networks [88].

A master can have up to 5 radios, each of which can handle a maximum of 8 devices. Thus a master is allowed to have up to 40 devices in its network. The system uses TDMA to connect with the devices. The data transfer between the master and the nodes occurs in a time interval called W-cycle, composed of three W-subcycle. Each W-subcycle lasts approximately 1.6ms and allows a half-duplex communication, as depicted in Figure 2.23. If the data transmission fails in the first W-subcycle, the nodes have the two following subcycles to retransmit the message. In this way, IOLW ensures high reliability and deterministic latency. When retransmission is not required, the nodes can use the following W-subcycle to send non-periodic data. When a radio of the master handles eight devices simultaneously, the uplink part of each W-subcycle is divided into eight intervals where the nodes can send a message of 12 bytes maximum. If the application requires a longer payload, a node can use two time intervals in the subcycle to send 25 bytes, but in this case, the master would not be able the manage eight devices with that radio. [88].

1. https://io-link.com/en/

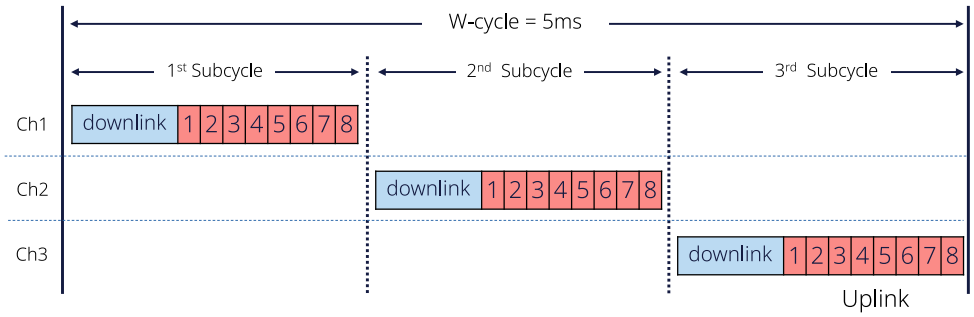


Figure 2.23 – Wireless IO Link W-cycle where the master and 8 slaves can exchange data, using the W-subcycle, and execute up to 2 retransmission when necessary [88].

2.2.4 Technologies Summary

In this section, we have presented four low-power network alternatives for the WBMS applications. They use the 2.4GHz ISM band with time-division (TDMA) scheme to avoid message collision and save energy. The Low Latency Deterministic Network (LLDN) MAC mode, defined in the IEEE Std. 802.15.4, uses the GACK method to increase the PDR. The main idea is to coordinate the nodes to allow one retransmission per message when necessary if there are enough retransmission timeslots. LLDN networks always use the same radio channel for the message exchange, so the network performance would be reduced if there is electromagnetic interference in this frequency interval. On the other hand, the IEEE Std. 802.15.4 TSCH MAC mode uses a different channel for each timeslot. The standard does not define how to schedule the message exchange and the retransmission within the network, so the network's reliability depends on the schedule defined by the upper layers. There are several proposals in the state of the art for the TSCH schedule definition. We have presented some examples of centralized and distributed scheduling algorithms. Each of them is designed with different objectives such as high reliability, low latency or dynamic response to the network architecture. However, when the wireless medium is polluted with external interference, the transmission's reliability is directly perturbed, even if the network uses channel hopping or an adequate scheduling function. In this section, we also presented several alternatives proposed in the state of the art that allow us to avoid the channels with bad quality (channel blacklisting), improving the network's overall performance. The IEEE Std. 802.15.4 proposes OQPSK modulation with 16 channels for the physical layer, allowing up to 250Kb/s of data rate. On the contrary, BLE networks use GFSK modulation with 40 channels and up to 2Mb/s.

	IEEE802.15.4 TSCH	**IEEE802.15.4 LLDN**	**BLE**	**Wireless IOL**
Physical layer	OQPSK	OQPSK	GFSK	GFSK
Max data rate	250 Kb/s	250 Kb/s	2 Mb/s	1 Mb/s
Frequency diversity	Yes	No	Yes	Yes
Time division	Yes	Yes	Yes	Yes
Max payload	125B	125B	250B	25B
Retransmission	Depends on the schedule	One per message GACK method	Unlimited	Up to 2

Table 2.1 – Comparison between low-power network candidates for the WBMS application. All of the presented alternatives uses the 2.4GHz ISM band.

This specification defines a central device that must establish a connection with some peripherals. When a message is lost, the protocol executes the necessary retransmission until success, ensuring high reliability. However, the retries occur in each node's assigned connection events, which increases the delivery latency. Wireless IO-Link allows managing up to eight devices with one radio interface. It guarantees 5ms of latency, including two retransmission opportunities to ensure high robustness. If the master needs to manage eight devices, the maximum payload length is limited to 12 bytes. It can be extended to 25 bytes, but the number of devices would be reduced to 4. In a WBMS application, the master would need multiple radio interfaces, leading to high energy consumption. Table 2.1 shows a comparison summary between the four alternatives.

We have seen that some properties of these protocols are not fully compatible with the required system. Like no frequency diversity in LLDN, OQPSK with only 16 channels for IEEE802.15.4 TSCH, no deterministic latency for BLE or limited payload and number of devices for IOLW. At the same time, they have some characteristics that give us ideas to design an appropriate WBMS network. We can conclude that it is mandatory to use frequency diversity together with time-division to gain robustness and reduce power consumption. Also, a centralize schedule function for the message exchange is more convenient as the WBMS has deterministic traffic. This schedule must allow multiple retransmissions within a specific time interval to increase reliability and enable bounded latency. Finally, a channel blacklisting method should be included as it would allow the network to choose the radio channels with the best quality and reduce the number of packets lost.

2.3 WBMS state of the art

Implementing such a challenging network has attracted the attention of both the academic and the industrial world. In this section, we first overview the academic works where the authors have developed a WBMS system using different types of networks. Then, we talk about the commercial solutions of WBMS available at the moment of writing this document. In particular, we focus on the systems provided by Texas Instrument (TI) and Analog Devices (ADI), which are the more mature WBMS propositions in the market.

2.3.1 Related work

WBMS has been an active research subject in recent years. Several authors propose different ways to implement it for stationary and automotive applications. The main question concerns the most appropriate wireless protocol and network topology for such a critical application. Huang et al. in [89], proposes the Wireless Smart Battery Management System (WSBMS), which uses a WiFi connection between the master and the slaves. It can also compute the SOC and SOH, keeping the battery cells balanced. This work does not provide details about the network performance (reliability) or the power consumption. It is well-known that WiFi networks are designed for high throughput without special considerations regarding bounded latency. Some other authors propose to use a Bluetooth Basic Rate (BBR) network. Jamaluddin et al. proposed a BBR-based wireless battery monitoring system that uses a Bluetooth module to share the battery data with a computer in a stationary application [90]. Shell et al. present a BBR WBMS system designed and implemented for a go-kart built for racing to reduce the mechanical failure of the wired BMS [91]. In both works the proposal only used one slave and one master. None of them present data about latency, reliability, or energy consumption. Though BBR consumes much less power than WiFi, it is still high for WBMS and can not warranty deterministic latency, which is crucial in a critical application [92].

Rahman et al. [93], Meyer et al. [94], and Wu et al. [95] propose a WBMS implementation based on the Zigbee protocol. The latter monitors a 50 Ah battery pack composed of 108 Li-cells with a sampling period of 200 ms. The authors affirm that the maximum current consumption of the devices is 28 mA. However, there are no network performance results in the analysis. Even though Zigbee is a standardized low-cost and low-power wireless protocol, it has several limitations for critical applications such as the WBMS. It is not designed to guarantee deterministic latency or high-reliability [92]. Kunitachi

et al. present a reliable wireless communication for WBMS based on a TSCH network. They propose using the overhearing technique to establish a highly reliable link between the nodes and the master. With this protocol, when a source node sends a packet to a destination, the neighbor nodes overhear the packet. If the destination does not receive the packet, the neighbor nodes transmit it again using CSMA to check if the channel is free. The network simulation results show that it can achieve up to 100 % reliability [96]. The article does not present energy consumption results, but the overhear technique requires high radio utilization, increasing the power consumption of the nodes. M. Lee et al. propose a WBMS based on a proprietary protocol called Wireless Battery Area Network (WiBaAN) [97] [98]. This protocol uses FSK modulation with up to 1Mbps of data rate in the 900MHz ISM band. It does not use channel hopping because they prefer to use channel diversity to allow multiple WBMS networks in the same space. The authors affirm achieving a reliable network, but no data from simulations or experimentation are presented.

Le Gall proposes a wireless network for reliable electric vehicle BMS in [2]. Using a star network with the BLE PHY layer, he executed some experiments in a battery pack to measure the link quality between wireless nodes. The results showed that with retransmissions, the packet delivery ratio was at least 98.5 % when there was no intentionally generated electromagnetic interference. It also mentions that WiFi interference significantly impacts the PDR, so the author proposes using channel blacklisting to mitigate external interference. He also highlighted that the network performance does not seem to be affected when the car is in driving mode. The prominent finding of these tests is that the link quality depends on the position of the nodes. For this reason, he proposes a method to implement a two-level tree topology based on the link quality to create a more robust wireless communication for the WBMS. In this topology, the nodes in rank two send a package to the nodes in rank one, aggregating all the messages received to send them to the master of the network using one packet. The wireless network was implemented and tested in a Renault Fluence battery pack environment using the IEEE Std. 802.15.4 TSCH physical and link layer. The final tests show that the network can adapt its topology based on the link quality, so wireless communication is feasible for the WBMS. However, he mentions that there are still some important aspects to consider before implementing in a real EV, like security problems, BMS sleep mode, cost advantage, power consumption, and topology definition.

2.3.2 Commercial WBMS

Important actors in the electronic and automotive world propose a WBMS solution. They all affirm that this breakthrough technology reduces the battery pack assembly and design complexity. Marelli has announced that its solution will be ready in the second quarter of 2022 to support customer launches in 2024 [99]. Renesas proposes a BLE-based solution using their SmartBond TINY™, which they affirm is the smallest and lowest power BLE SoC in the world [100]. Analog Devices proposes a WBMS solution now used in the Hummer EV released in 2021 by General Motors (GM) [9]. Texas Instrument has launched the first WBMS TUV SUD assessed for enabling ASIL D functional safety systems [101]. In this section, we present some details of the two last examples, which are the most mature in the market.

Texas Instruments WBMS

Texas Instruments (TI) affirms that cable failures into a BMS are costly, the leading failures occur with the connectors and wiring harness. They claim that implementing a wireless connection reduces weight, saves costs and avoids typical isolation problems of traditional wired BMS [101]. Based on their well known Simplelink MCU platform, which is a low power wireless connectivity solution for the IoT, TI has created a WBMS commercial implementation [102]. They have developed a proprietary protocol for their WBMS solution, which uses the 2.4GHz band with the frequency hopping technique and 2Mbps of data rate. The vendor affirms that their system achieves 99.999 % of reliability [103] but, as it is a proprietary protocol, there are no public details of how the network manages the nodes to achieve this announced robustness. Their proposal is the first WBMS TUV SUD assessed for enabling ASIL D functional safety systems [101], to achieve this TI uses the black channel concept. There are two possible architectures for safety-relevant data transmission; white channel and black channel. In the first, all the hardware and software components are developed and validated according to functional safety standards. On the other hand, the black channel concept means that the end devices (BMS controller and ASIC BMS) must be safety compliant but it is not necessary for the components of the communication channel, which is the case of the wireless MCU used in the system [104]. The TI WBMS system features CRC-32 to detect data corruptions and four-byte MAC (Message Authentication Code) to ensure authenticity and integrity. The wireless protocol encapsulates the BMS ASIC commands and responses without modifying the

data. In this way the system can detect errors in the communication channel without interfering with the safe communication protocol between the BMS controller and the ASIC BMS [104].

Analog Devices WBMS

SmartMesh IP is the commercial TSCH solution of Analog Devices (ADI) which announce an end-to-end reliability of 99.999 % [105]. The manufacturer created a microcontroller with a hardware MAC engine to execute the MAC related operations at a lower power consumption than using a microcontroller [106]. Analog Devices (ADI) has seen the need of the industry to implement robust wireless communications inside a battery pack, so they propose a WBMS system based on a modified version of the proven SmartMesh IP technology. The first ADI WBMS concept was implemented in a BMW i3, the main motivation was to improve the system reliability, reduce the cost and wiring complexity of the battery pack. ADI mentions that the timing synchronization characteristics of a TSCH network helps the BMS algorithms to compute more accurately the SOC and SOH. Additionally, they foresee that with the additional local processing at each module there is a potential to enable the smart battery modules concept [107]. In december 2021,General Motors (GM) released the first commercial EV with a WBMS from ADI, the Hummer EV. ADI claims that their WBMS system facilitates the diagnostic of malfunctioning modules, their replacement and their integration into new second life applications [9]. The system has achieved the ISO/SAE 21434 certification, which is an assessed automotive cybersecurity certificate [108]. To the best of our knowledge, there are no available details to the public of how their protocol manages the network to achieve high reliability.

2.4 Conclusion

In this chapter, we have discussed several elements to understand the context and challenges that involve the implementation of a WBMS. First, we have shown the basic architecture of an EV where we have detailed how the battery pack interacts with multiple elements in the vehicle and the electromagnetic interference generated by the power train. Then we presented the basis of lithium-based cells, the most common receptacle for these systems. Although these batteries have several advantages, which have made possible the massification of EVs, Li-ion cells represent a significant challenge in building safety-critical systems. When the voltage or temperature of these cells is out of the safety window, the

battery's lifetime may be considerably reduced or even provoke catastrophic consequences. To avoid these undesirable effects, the battery pack includes a BMS to ensure optimal operation conditions. We have defined the BMS as a set of voltage, temperature, and current sensors that collect data to estimate the State of Charge (SOC) and Health (SOH) of the cells. With this information, the BMS can allow/disallow the charge and discharge, set the maximum input/output power limit, and alert the exceeding of the safety window. In this chapter, we presented the classical architecture of the BMS, which uses a wired interface between the sensors (slaves) and a computer unit (master). We also reviewed this critical system's basic algorithms and introduced the notion of battery pack balancing. In this context, replacing the wired interface between the slaves and the master with a wireless link represents a significant challenge but also brings several advantages.

In order to achieve a similar level of robustness using a wireless interface in the BMS, it is necessary to implement an appropriate protocol. This implementation must provide a highly reliable link with bounded latency and low power consumption. The next part of this chapter was focused on the wireless network protocols that may fulfill all the requirements of a WBMS. We presented the IEEE Std. 802.15.4, which is the standard for Low Rate Wireless Personal Area Network (LR-WPAN). In particular, we reviewed the LLDN and TSCH MAC modes proposed in this document. The first one uses time division with the GACK mechanism to achieve high reliability and bounded latency. The second uses time division and frequency diversity to gain robustness against external interference. Both use the same physical layer with OQPSK modulation and 250kb/s of data rate. Then, we presented the BLE networks, which use time division and frequency diversity with GFSK modulation at 2Mb/s. These networks present high reliability as they execute the necessary retransmission until successful delivery, but it does not ensure a specific latency. We also reviewed Wireless IO-Link, which provides bounded latency for small networks. Each master supports four slaves with payloads of 25 bytes or up to 8 slaves with a smaller message length. In this chapter, we show that even if these networks present several advantages, none of them can guarantee all the particular requirements of a WBMS application. We also learned that time division, frequency diversity, channel blacklisting and good retransmission management are the keys to create an appropriate network to replace a wired interface in a critical system like the BMS.

In the last part of this chapter, we reviewed the state-of-the-art of WBMS. We first reviewed some academic works that have used several kinds of networks to validate the viability of introducing a wireless link in this type of system. Most of these examples do not

present detailed performance data of the systems or a complete implementation of an EV. Then, we talk about commercial WBMS options available in the market when writing this document. In particular, the solutions proposed by TI and ADI are the most mature. Both solutions affirm to provide 99.999 % of reliability, but none of them show how they manage the network to achieve this level of performance. We have seen in this chapter that a special wireless network should be implemented for such a critical system because standard networks can not guarantee the particular requirements of the WBMS application. The following sections of this document describe our protocol stack proposition for wireless critical applications, like the BMS, which should ensure high reliability, bounded latency, and low power consumption. Then, we describe how we have implemented and tested our proposal in a Renault Zoe battery pack. Once the wireless protocol has been validated, we present the future of the WBMS and how the new hardware used in this application has the potential to decentralize the BMS algorithms.

Chapter 3

Robust wireless communication for BMS

In the previous chapter, we presented the technical background to understand the challenges of replacing the wired connection within the BMS with a wireless link. After reviewing three of the most promising technologies for this application, we observed that none of them was designed to meet the requirements of critical applications. For this reason, this chapter presents our proposed wireless network protocol that achieves low power consumption, bounded latency, and high reliability. This proposal retakes concepts of the protocols reviewed in chapter 2, like GACK retransmission, time division, frequency diversity and channel blacklisting. Section 3.1 describes how we use these elements to create a generic framework that can be adapted to multiple applications to enable wireless communications in critical environments. We detail our network's scheduling function for the MAC layer, the chosen physical layer, the new timeslot architecture and our proposed blacklisting algorithm. Section 3.2 details the adaptation process of our framework to the Renault Zoe WBMS application and how we implement it in real hardware. Section 3.3 presents the results of some tests we executed to validate our proposal. First, we compare our method with a simple retransmission algorithm in an environment with generated electromagnetic interference. Then, we test our system in a Renault Zoe battery pack for several days to measure the network reliability. Next, we measure the energy consumption of the nodes to prove that our proposal meets all the requirements of the WBMS. Last, we probe the advantages of including the blacklisting algorithm in an environment with high external interference. Finally, in section 3.4 we summarize the chapter, present some conclusions and discuss the next steps of our work in order to achieve a complete WBMS solution.

3.1 Wireless communication for critical applications

Wireless networks are widely used in the industry to facilitate the deployment of sensors and actuators. Typically, these WSNs have a central node that periodically receives data from sensors and sends commands to the actuators. However, when it comes to critical applications where a failure in communication could end up in catastrophic consequences, the industry prefers to use a wired link between the central devices and the peripherals. In order to allow the inclusion of wireless communication in critical WSN applications, we have developed a framework that can be adapted to multiple applications requiring high reliability, bounded latency, and low power consumption. In this work, we propose a TSCH network using a centralized GACK-based schedule with an adapted channel blacklisting algorithm. Our network uses a star topology running over the BLE physical layer. The star topology guarantees a similar energy consumption of the nodes, while the GACK method allows dynamic retransmission management. The TSCH MAC protocol takes advantage of time division for collision avoidance and radio channel hopping to mitigate external interference. The BLE physical layer allows us to achieve higher throughput and shorter timeslot duration. Figure 3.1 presents an overview of the proposed protocol stack, whose operation principle will be explained throughout this section.

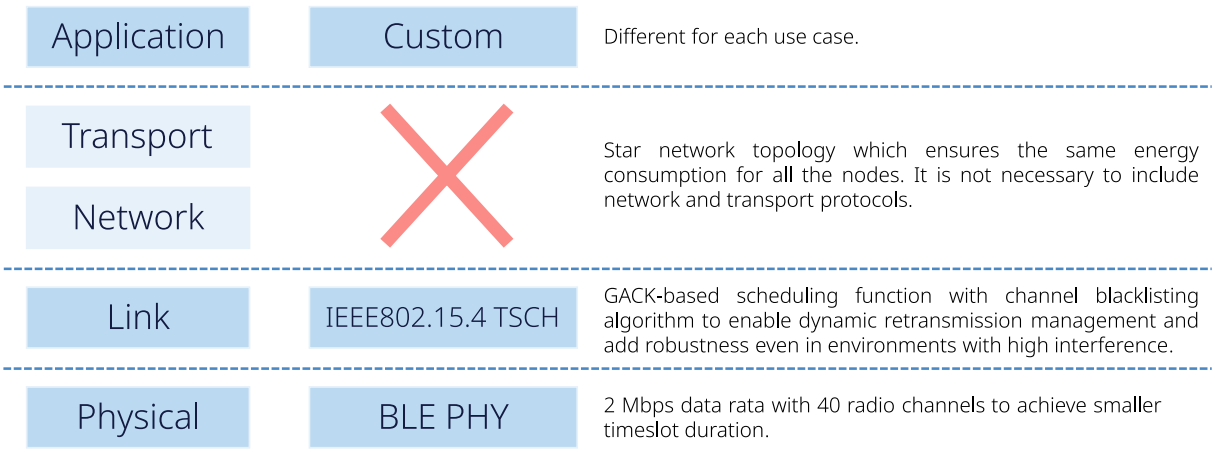


Figure 3.1 – Proposed WBMS network protocol stack to enable high reliable low power communication in critical applications.

Figure 3.2 presents our scheduling function proposal for a WSN composed of a central device and N nodes where the root must receive a new packet from the peripherals every Target Refresh Time (TRT) milliseconds. The root is responsible for assigning a unique

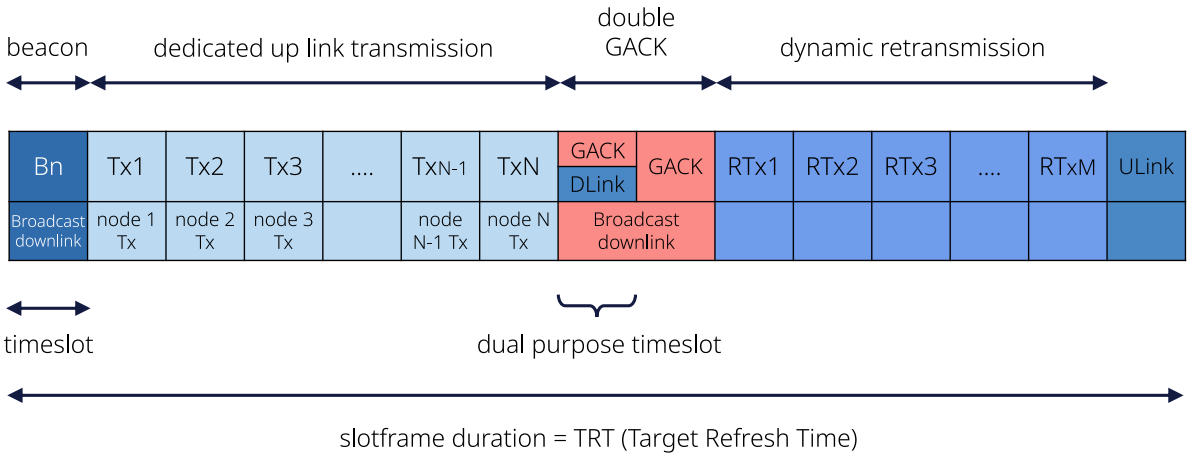


Figure 3.2 – Proposed TSCH centralized scheduling function for critical applications. The sloframe has a static part with dedicated timeslots for the N nodes of the network and a dynamic section to allocate the retransmission timeslots according to the needs of the network.

identifier (ID from 1 to N) within the network to each node during the joining phase. Our scheduling function is divided into two parts, the first one for static transmission and the second for dynamic retransmission. The first timeslot is reserved for the beacon frame to maintain synchronization and allow new nodes to identify and join the network. Then, we have N uplink timeslots for each slave node where the individual acknowledgment is not required as we use the GACK mechanism. These timeslots are distributed according to the node ID, so the node with the ID equal to 1 uses the first uplink timeslot. Since wireless communication is naturally unstable, the number of retransmissions required in each slotframe is variable. To efficiently manage the retransmission timeslots distribution, we use duplicate GACK frames to inform the slaves about the messages that were not received by the root. We have chosen this approach to minimize the probability of losing a GACK frame, as the root can send it two times over different radio channels. The GACK information is sent in a bitmap format where each bit is dedicated to informing one specific node based on its ID. If the corresponding bit is set to 1, the message is lost, and retransmission is needed (negative acknowledgment). If the message was successfully delivered, the node goes to “sleep” mode until the next slotframe. Based on the GACK frame, the nodes can derive which retransmission timeslot they can use to avoid message collision within the network. More details on the dynamic retransmission management are given in the next section. Our proposed scheduling function uses two management timeslots (ULink and DLink) for the joining process. When a new node wants to join the

network, it listens for an EB to synchronize its clock and sends a joining request to the root using the ULink timeslot. The root sends the joining response with the assigned ID to the node using the timeslot N+1 (DLink). This timeslot has dual purposes; when there is no joining response message from the root on the queue, it is used to send a GACK. Note that when the nodes in the network receive the joining response sent for the new node, they can detect in the header that it is not a GACK message, ignore it, and wait for the GACK in the next timeslot (N+2). As you can observe, if a node loses synchronization, it can rapidly rejoin the network without interrupting the operation of the other slaves.

3.1.1 Dynamic retransmission

A common approach to executing retransmissions is to allocate dedicated timeslots to each node to retransmit their message. For our proposal, we have chosen to use a GACK-based dynamic retransmission method rather than a classic static method for the following reasons. First, reliability is an important requirement for WSN in critical applications. This mechanism allows the network to allocate the available retransmission timeslots to the nodes that need to retransmit their message, giving them multiple retransmission opportunities. In a static method, the slaves have a fixed number of retransmission opportunities, and if they do not need to use their assigned timeslot, it cannot be reassigned to give more retries to the other nodes. The main objective is to create a network that can adapt rapidly to variable external conditions. The second reason is that WSN nodes are generally battery-powered, and low energy consumption is essential. Dynamic retransmission also allows the nodes to send their data only when required, thus, they can go to "sleep" mode when their task is completed.

In order to explain how our proposed dynamic retransmission method works, we present in Figure 3.3 four possible retransmission distribution examples of a network composed of 10 nodes. As you can observe, the scheduling function has ten timeslots for the retries. Figure 3.3a presents the first case where the root informs the network in the GACK frame that all the messages were received. In this situation, all the nodes go to sleep mode until the next slotframe. In the second case, Figure 3.3b, the root tells the nodes that the messages from nodes 3, 5, and 8 are missing. In this case, the three slave nodes retransmit their message using the first three timeslots while the rest of the nodes go to sleep mode until the next slotframe. The node with the lowest ID (3) uses the first retransmission timeslot (RTx1), the following node ID (5) uses the second one (RTx2), and so on. If the root successfully receives the three missing frames, it will inform the

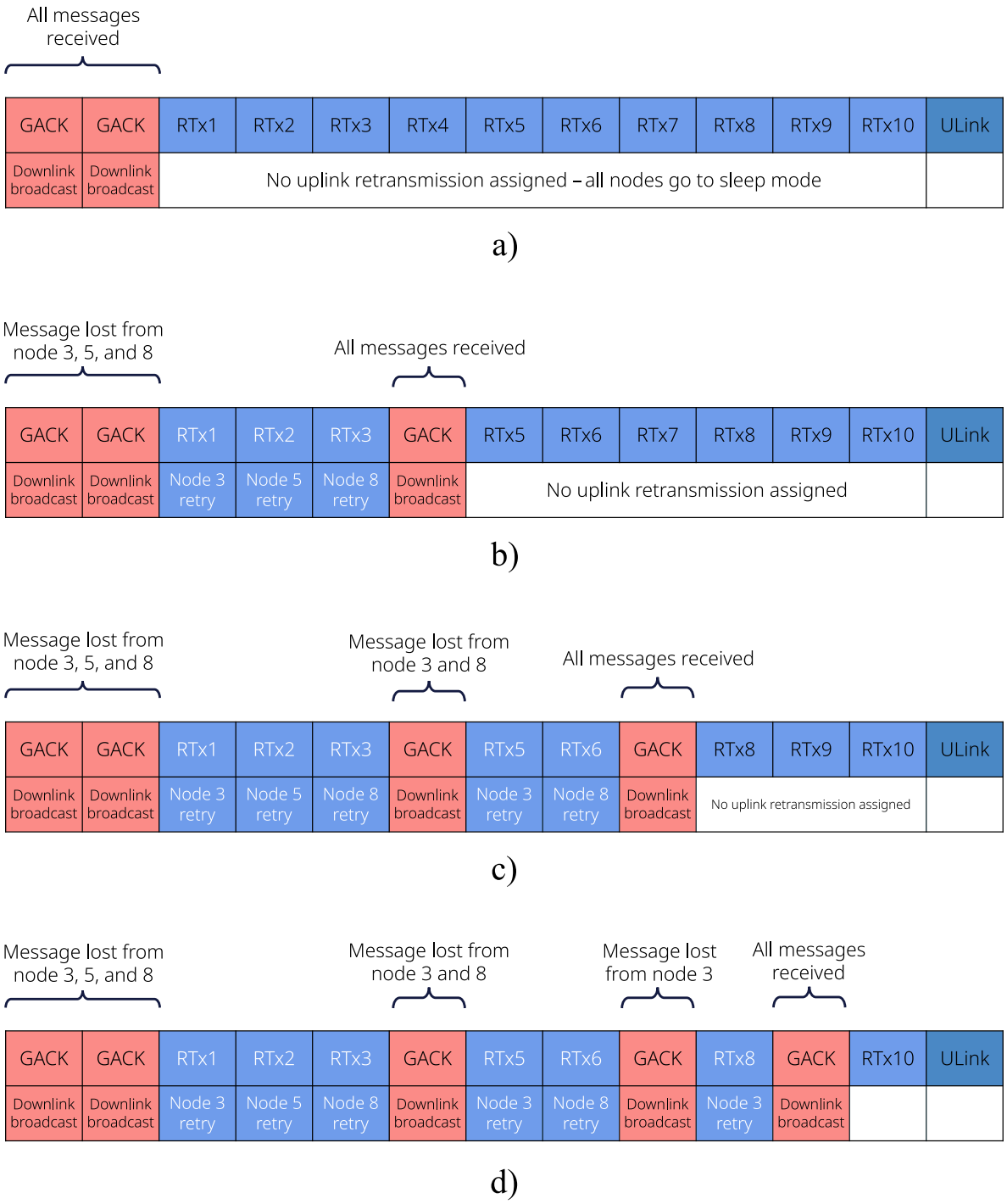


Figure 3.3 – GACK based dynamic retranmission management examples. a) All the messages were received by the root. b) Messages from nodes 3, 5 and 8 are lost and need to be retransmitted. c) After one retransmission, the messages from nodes 3 and 8 are still lost. d) After two retransmission the master has not yet received the message from node 3. The dynamic approach allow the network to give multiple retransmission opportunities to the nodes without internal collisions.

network allowing nodes 3, 5, and 8 to go to sleep mode. On the contrary, the process can be repeated if some messages are still lost. In Figure 3.3c, after the first three retransmissions, the root tells the nodes that messages from nodes 3 and 8 are still missing. With this information, node 5 can go to sleep mode until the next slotframe, while nodes 3 and 8 use the following two timeslots to execute another retry. The process can continue until the end of the slotframe or until the root has received all the messages. As shown in Figure 3.3d, if the message from node 3 is still missing, it can use the RTx8 to retransmit its message and, eventually, the RTx10. Note that in the worst case, where all the messages from the network are lost, each device can have one retransmission opportunity, as in a static retransmission approach. If we use a static method in the situation of Figure 3.3c, nodes 3, 5, and 8 would have only one retransmission opportunity, and the messages from nodes 3 and 8 would be definitely lost, which reduces the reliability of the wireless network. The maximum number of retransmission opportunities is a configurable parameter that might vary from one application to another. It depends on several factors, such as the possible number of timeslots in a slotframe, the required reliability or the interference level. Note that there is a trade-off between network reliability and power consumption; the higher the number of retransmission opportunities, the higher the energy consumption.

3.1.2 Physical layer and timeslot architecture

The IEEE Std 802.14.5 establishes a timeslot length of 10ms for the TSCH MAC mode when the nodes operate in the 2.4GHz band. It states that it is mandatory to use an OQPSK modulation with a 250 kb/s data rate on the physical layer, as explained in Section 2.2.1. Moreover, it specifies that the 2.4GHz band must be divided into 16 radio channels where adjacent frequencies are 5 Mhz apart. However, a timeslot of 10 ms may lead to a large slotframe size if several nodes need to send their data periodically. A large slotframe would not meet the TRT application requirement for some critical applications. Also, the number of radio channels means that we can have up to 16 simultaneous transmissions, which may not be sufficient for certain industrial environments. To overcome these issues, we propose to employ the TSCH MAC layer over the Bluetooth Low Energy (BLE) physical layer, controlled by our proposed dynamic retransmission schedule. The BLE PHY operates in the 2.4GHz band. It divides the band into 40 radio channels and uses the GFSK modulation at 2 Mb/s, allowing us to exchange one frame in a shorter time.

To reduce the 10 ms timeslot length proposed by the IEEE Std 802.15.4 using the

BLE PHY, there are two points to consider; the maximum payload to be sent over the wireless medium and the necessary time for post-processing that the hardware requires at the end of each timeslot. In Figure 2.15, the IEEE Std. 802.15.4TSCH timeslot structure is depicted. In our case, we do not have to execute the CCA mechanism at the beginning because all the timeslots are reserved for dedicated nodes. Thus, the Tx and Rx offset values can be reduced to the minimum. These minimum values depend on the time required by the hardware to wake up and initialize all the necessary peripherals. Thanks to the GACK mechanism in our schedule, we do not need the specific time interval for the ACK within a timeslot, as it is usually done. Therefore, we can delete the ACK field at the end of the timeslots. In Figure 3.4, our proposed timeslot structure is illustrated. The first field (RxOffset) is the time the microcontroller requires to be ready to receive a new message. Next, we have the Tx/Rx interval where the nodes execute the data frame transmission. Finally, there is a dedicated interval for real-time post-processing, which depends on the number of calculations and the processing capacity of the nodes. In Section 3.2.1, we will empirically define the length values of each part of the timeslot for the hardware platform used in our WBMS implementation.

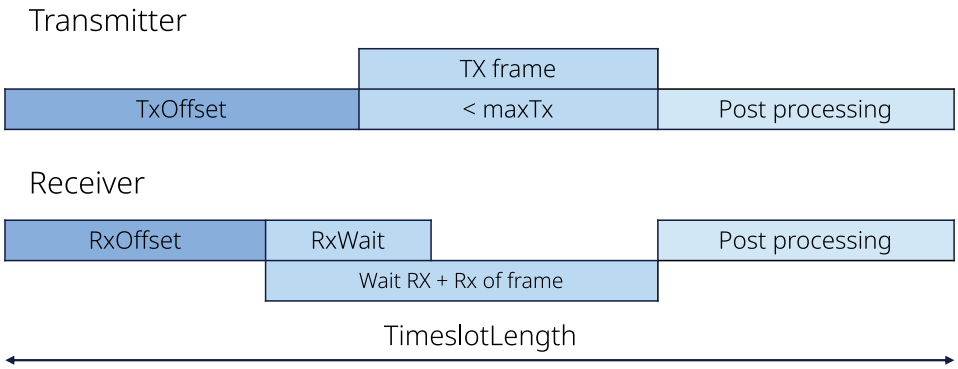


Figure 3.4 – Proposed timeslot architecture for wireless communication in critical applications.

3.1.3 Channel blacklisting

As mentioned in Section 2.2.1, using channel hopping and retransmission mechanisms to increase the performance of a TSCH network would not be enough if there is high electromagnetic interference. The network should be able to identify the best radio channels and avoid the bad ones to achieve the required level of reliability in critical applications.

We have chosen to reuse some concepts reviewed in state of the art to propose a channel blacklisting method adapted to our scheduling function, detailed in Algorithm 1. We use a similar approach of LABeL to identify the bad channels. The root computes periodically (every T_{update}) the stability of each channel based on the PDR and using the WMEWMA estimator [80]. During this period, the root counts the number of packets received, how many were lost, and which channel was used in the dedicated timeslots. Then, it updates the channel stability using equation 3.1 and creates the new channel blacklist using the best channel as a reference. As in the LABeL case, we propose to set a minimum limit for the number of good channels. If the initial threshold does not return enough good channels, the algorithm can iterate, reducing the threshold until it obtains the desired number of acceptable channels. Note that it is essential that the number of timeslots in a Slotframe is not a multiple of the number of channels. Otherwise, each timeslot would always use the same radio channel, and it would be impossible to compute the PDR using only the dedicated timeslots. As presented in Section 2.2.1 the α factor determines the dynamic of the estimation, if the network is in a environment with transitory interference it is better to use a small factor.

$$PDR_{wmewma}[ch] = \alpha * PDR_{wmewma}[ch] + (1 - \alpha)PDR_{T_{update}} \tag{3.1}$$

Once the root has a new channel blacklist, it must share it with the slaves. To avoid channel list mismatch within the network, we propose using the method presented in [79], where the root shares the channel blacklist with a target ASN (ASN_{blk}) from which the nodes should start using the new sequence. We propose giving at least a ten slotframes window for the blacklist delivery before the network starts using it. To ensure that all the nodes receive the new sequence information, we share it by including this data in the GACK frame, which is sent at least two times within a Slotframe in different channels. Once the ASN_{blk} is reached, the nodes must build a new channel hopping sequence from the blacklist and start using it, as depicted in Figure 3.5. In our algorithm, the node must first build an ordered list of the good channels. Then, it creates the new sequence using the default channel list as a reference. Each time it finds a bad channel, the nodes replace it consecutively with the channels of the good list. This method has two advantages; the first one is that the new sequence keeps backward compatibility with the location of the good channels in the default sequence. If one node does not receive the new blacklist for any reason, it will not lose the connection and eventually will receive the new list. The

second advantage is that if two consecutive channels in the sequence must be replaced, the algorithm assigns a different good channel to each location. In this way, we keep the well-known characteristics of channel hopping in our new sequence.

To check if a blacklisted channel can be recovered and used again by the network, we propose to set three levels of blacklisting. The idea is that the first time a channel is included in the blacklist, it enters the first level with a banning time of T_{lev1}. After the banning time has finished, it is excluded from the blacklist and tested during T_{update}. If it is found again as a bad channel at the end of the test period, it will be blacklisted and enters the second level, where the duration is T_{lev2} ($T_{lev2} = 5 * T_{lev1}$). Finally, if it is tested again after the banning time and is still a channel with low stability, it will enter the third blacklisting level, where the banning time is T_{lev3} ($T_{lev3} = 10 * T_{lev1}$). This way, the system is robust against channels with transitory interference (level 1) and channels with permanent interference (level 3). As we propose a system for critical applications, the algorithm only allows picking one bad channel (reintegration channel) for every T_{update} that has finished the ban and testing it. In order to not affect the high reliability required by these applications, the nodes can only use the reintegration channels during the dedicated uplink timeslots. If the channel sequence indicates to use this radio channel for a GACK or a retransmission frame, the nodes must use the channel with the highest stability instead.

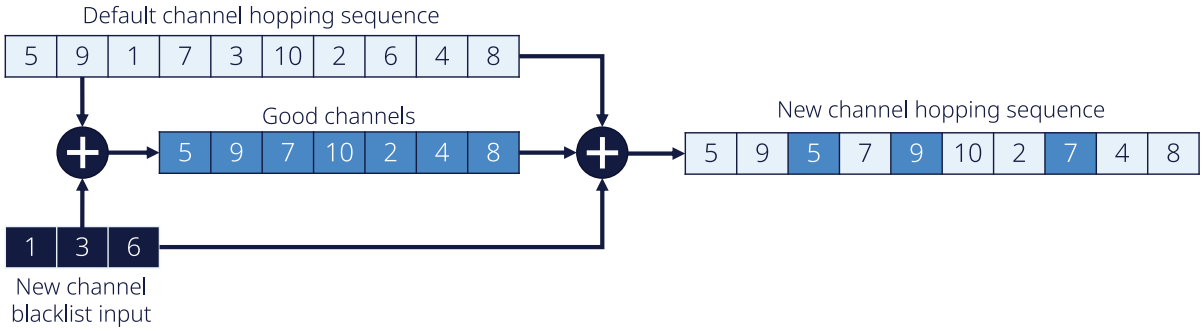


Figure 3.5 – Channel sequence generation based on the default sequence upon arrival of a new channel black list. Note that the good channels keep their original position and we just replace the bad ones.

Our proposal needs the master to use the GACK frames to share with the network the future blacklist, the target ASN_{blk}, the reintegration channel, and the channel with the highest stability. In order to allow new nodes to join the network, the master must also include this information in the beacon with the current blacklist used by the nodes.

Algorithm 1 : Channel blacklisting algorithm

while True **do** ▷ Every T_{update}

/*** Compute channel stability and find the best one ***/
for c $\notin$ *blackChannels* **do**
$PDR_{new}[c] \leftarrow nMsg_{received}[c]$ / $nMsg_{expected}[c]$ ▷ Compute PDR
$PDR_{ch}[c] \leftarrow \alpha * PDR_{ch}[c] + (1 - \alpha) * PDR_{new}[c]$ ▷ WMEWMA
if $PDR_{ch}[c] > best$ **then** ▷ Find the best channel
$best \leftarrow PDR_{ch}[c]$
end if
end for
$minPdr \leftarrow best - \beta$ ▷ β can be adapted according to the application

/***Create channel blacklist ***/
while $goodChannels > minCh$ **do** ▷ Ensure at least $minCh$ good channels
$goodChannels \leftarrow 0$
for c $\notin$ *blackChannels* **do**
if $PDR_{ch}[c] < minPdr$ **then**
$blackChannels = blackChannels + \{c\}$ ▷ Add to black channel list
$banDuration[c] \leftarrow getBlacklistingLevel(c)$
else
$goodChannels \leftarrow goodChannels + 1$
end if
end for
for c $\in$ *blackChannels* **do**
if $banDuration[c] == 0$ **then**
$blackChannels \leftarrow blackChannels - \{c\}$ ▷ Remove from list
$reintegrationChannel \leftarrow c$
break ▷ Only one reintegration channel allowed at the same time
end if
end for
$minPdr \leftarrow minPdr - \beta$ ▷ Reduce threshold if necessary
end while
end while

3.2 Renault Zoe WBMS

WBMS is the target application for our proposed TSCH-based schedule with dynamic retransmission. In this section, we present how we designed and implemented a robust wireless network for the active mode of the WBMS of a Renault Zoe to validate our framework proposal described in the previous section.

The Renault Zoe battery pack has 12 modules, each containing eight cells. Thus, our network should be composed of a root, the WBMS master, and 12 nodes, the WBMS slaves. In active mode, the WBMS of the Renault Zoe has the following requirements:

- The root must receive all the 96 voltage measurements every 100 ms.
- The number of data packets successfully delivered to the root (WBMS master) must be at least 99.999 %.
- The average current consumption of each WBMS slave must be less than 1 mA.
- The root must be able to send every 1 second a broadcast message to the slaves with the cell balancing commands.
- The energy consumption of the WBMS slaves must be similar to avoid cell balancing problems within the pack.
- If a node loses synchronization for some reason, it should be able to rejoin the network without interrupting the operation of the other slaves.

It is straightforward that if we keep the timeslot length of 10 ms as proposed by the IEEE Std 802.15.4 standard, the size of our slotframe will exceed 130 ms (beacon plus the dedicated timeslots) without even considering the retransmissions, which would not be enough to fulfill the reliability and latency requirements. Therefore, before defining the schedule, it is necessary to choose a hardware and software platform to specify the timeslot length and, thus, the number of timeslots we can have in a slotframe of 100 ms size.

3.2.1 Hardware and software considerations

The selected hardware for the nodes and the root is the Texas Instrument (TI) cc26x2 microcontroller. This chip has an ARM M4F as the main microprocessor, a radio transceiver controlled by an ARM M0 and the necessary timers to maintain the tight synchronization required in TSCH networks [109]. This platform is designed for low-power wireless applications with several power consumption levels: active mode for full functionality, idle mode to switch off the main CPU but keeping all the peripherals available, and

standby mode, which only keeps the I/O pins available to receive a wake-up order. For the first tests, the network was implemented using 13 TI Simplelink cc26x2 Launchpad (1 root and 12 slaves), an evaluation board for the cc2652 microcontroller, presented in Figure 3.6.

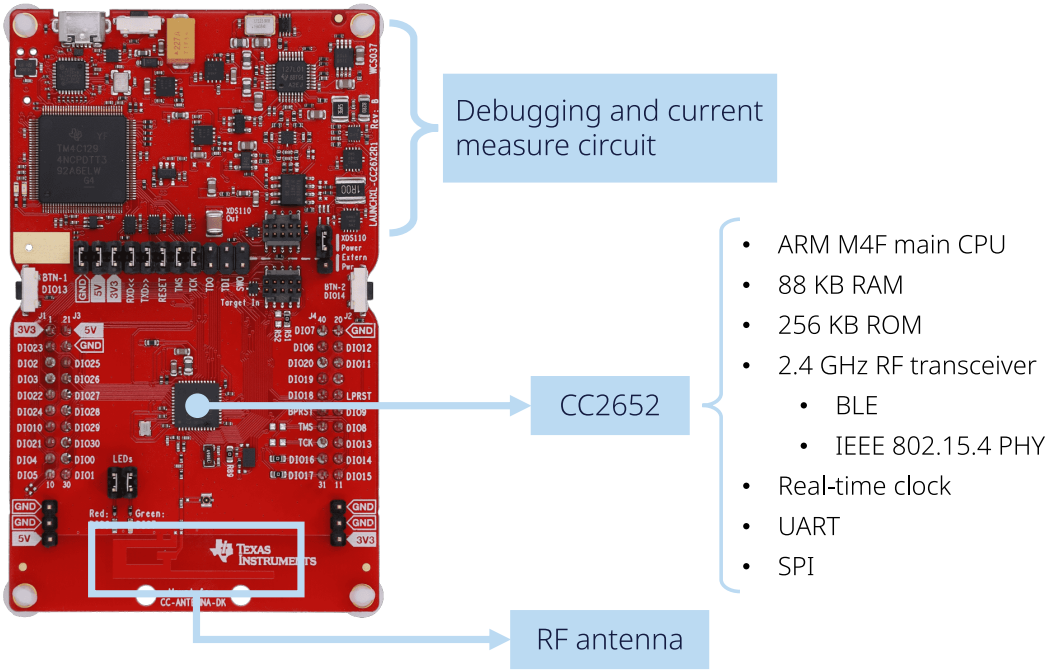


Figure 3.6 – Texas Instrument cc26x2 launchpad [110].

For the experiments on the Renault Zoe battery pack, the launchpad nodes were replaced by a custom board (WBMS slave) provided by TI which can be plugged to the battery modules interface. This board is principally composed of a cc2642 microcontroller for the network management and a BMS ASIC (TI bq79616) in charge of the battery cell measurements, as depicted in Figure 3.7a. Both devices communicate using an Universal Asynchronous Receiver Transmitter (UART) interface. The TI BQ79616 is an ASIL D-capable ASIC which measures up to 16 cell-voltage values with a precision of $\pm 1\ mV$ [111]. The custom WBMS slave board shares the same shape of the traditional Renault BMS slaves (Figure 3.7b) which let us use the same protection boxes for the wireless slaves (Figure 3.7c). This advantage allows an easy connection of the new slaves to the Renault Zoe battery modules, as depicted in Figure 3.8. Regarding the software, we were allowed to use and modify a proprietary implementation of the IEEE Std 802.15.4 provided by

Texas Instruments. We proceeded with this choice instead of the well-known open source implementations such as Contiki-NG or OpenWSN because these tools do not provide full support for the TI cc26x2 platform.

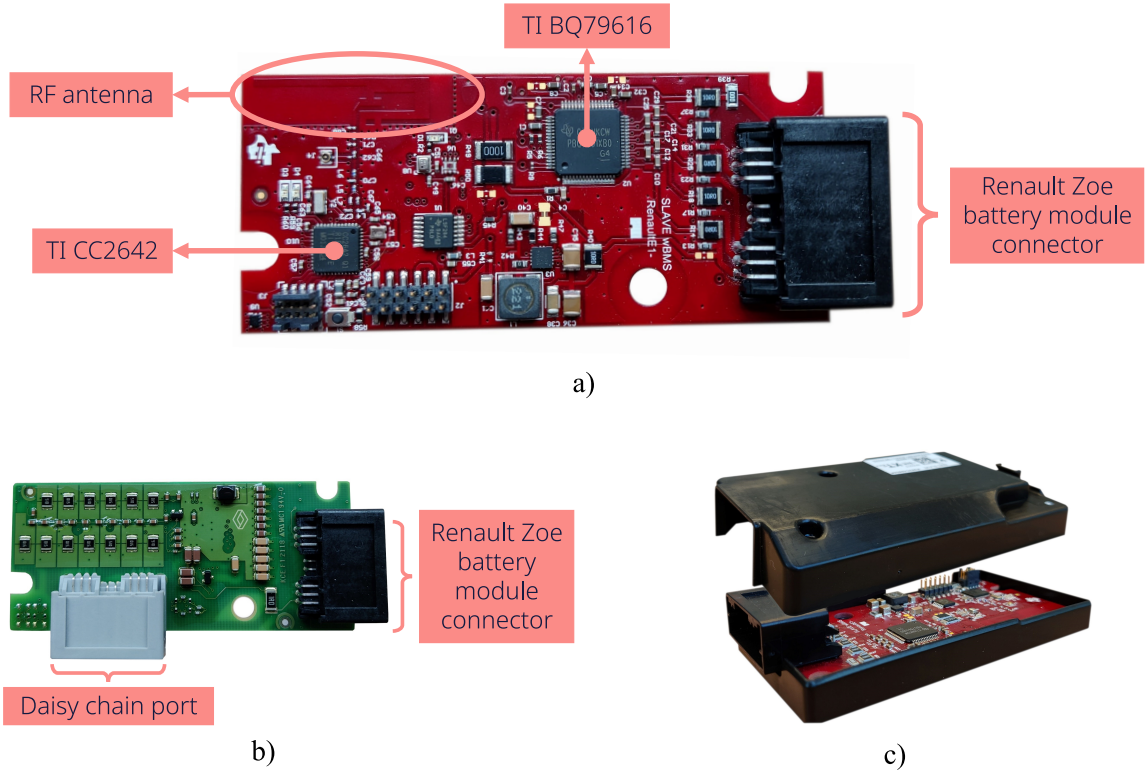


Figure 3.7 – a) WBMS slave provided by Texas Instrument which has the same shape as the Renault Zoe wired BMS slave. b) Renault Zoe wired BMS slave. c) The TI WBMS slave can use the same box of the wired version.

The first step of the implementation was to define the timeslot length for this platform. Toward this aim, we set a simple schedule with a slotframe composed of 13 timeslots, the first for the beacon and the following 12 dedicated for each node. Then, we executed several experiments changing the total length of the timeslot and Tx/Rx offset value. We wanted to observe in which cases the nodes were able to maintain synchronization and have sufficient time to finish the necessary tasks, for the timeslot operation, before the start of the next timeslot. Table 3.1 lists the timeslot's final parameter values, allowing us to achieve a length of 3300us.

For the Rx offset value choice, it is important to leave the necessary time for the device to initialize its peripherals when it goes from standby to active mode. If this interval is too

Figure 3.8 – Renault Zoe battery module connected to a TI WBMS slave which is inside the default Renault BMS slave box.

small, the node cannot receive data, and the network's performance will not be optimal. In our implementation, if a node has something to do in two consecutive timeslots, it cannot go to standby mode. After the first timeslot, it can only switch off the main CPU (idle mode). On the contrary, if it has enough free time to go to standby mode, it needs approximately 900 μs to recover full functionality, according to the manufacturer. Leaving this amount of time at the beginning of the timeslot would increase its total length, so we prefer to wake up the node 600 μs before the start of the timeslot and set the Rx Offset to 300 μs. It is possible to set a larger value for the Rx offset, but it would impact the final length of the timeslot or reduce the time interval for the post-processing. It may be possible to wake-up the node more than 600 μs before the timeslot start and reduce the RxOffset but, in our tests, this produced an abnormal behaviour of the network where the nodes constantly lose synchronization.

Regarding the guard time, which is the difference between Rx and Tx Offset, we configured it to 300 μs considering a timer frequency drift of 40 ppm, so in the worst case, the difference between the clock of the two nodes would be 80 ppm [71]. In this scenario, the nodes need to exchange a beacon message every 3,75 seconds to avoid losing synchronization, which gives us a good margin as the data exchange takes place every 100 ms. At this point it may appear that we can use a smaller value, but this choice was made considering the sleep mode which is described in the next chapter. Our solution allows a maximum payload size of 255 bytes, which means that by using the BLE PHY

	Length values (μs)
Rx Offset	300
Tx Offset	600
Max Tx frame	1020
Post processing	1680
Timeslot length	3300

Table 3.1 – Timeslot sections length distribution.

layer at 2 Mbps, we would need 1020 μs to send a full-size frame. Finally, the post-processing time in our case was set to 1680 μs; when we use a shorter time, sometimes the root was unable to finish its tasks before the end of the timeslot and missed the beginning of the next one. As the root has a defined task for multiple consecutive timeslots in our schedule, this behavior directly impacts the network reliability. Note that these values are completely dependent on the platform implementation and may vary if we use another software or hardware.

3.2.2 Dynamic schedule definition

Considering the timeslot length of 3.3 ms defined in the previous section, in a slotframe of 100ms, we can have 30 timeslots. In our proposed solution, we allocate the first timeslot for the beacon frame and 12 dedicated uplink timeslots, one for each WBMS node. Then, we have two timeslots for the GACK, one of which is also used for the joining responses from the root. Next, we have 14 timeslots for the dynamic retransmission mechanism and the last timeslot for the nodes' joining requests, as illustrated in Figure 3.9a. For this application, we did not set the maximum number of retransmissions a frame can have within a slotframe so only the available timeslots will limit it. As it can be observed, even in the worst-case scenario where all the message transmissions fail in the first 12 dedicated timeslots, the proposed schedule can support a retransmission round for all nodes within the same slotframe. On the other hand, in the best case, the nodes would deliver the messages in the first 12 timeslots, and then they can go to low energy consumption mode during the rest of the slotframe. Moreover, if any message transmission fails, it can be reattempted more than once to ensure high reliability. Regarding the downlink traffic, we have chosen to send the cell balancing orders in the same GACK frames, taking advantage of the large payloads. Another option would be temporarily allocating one of

the retransmission timeslots using the GACK message to inform the nodes.

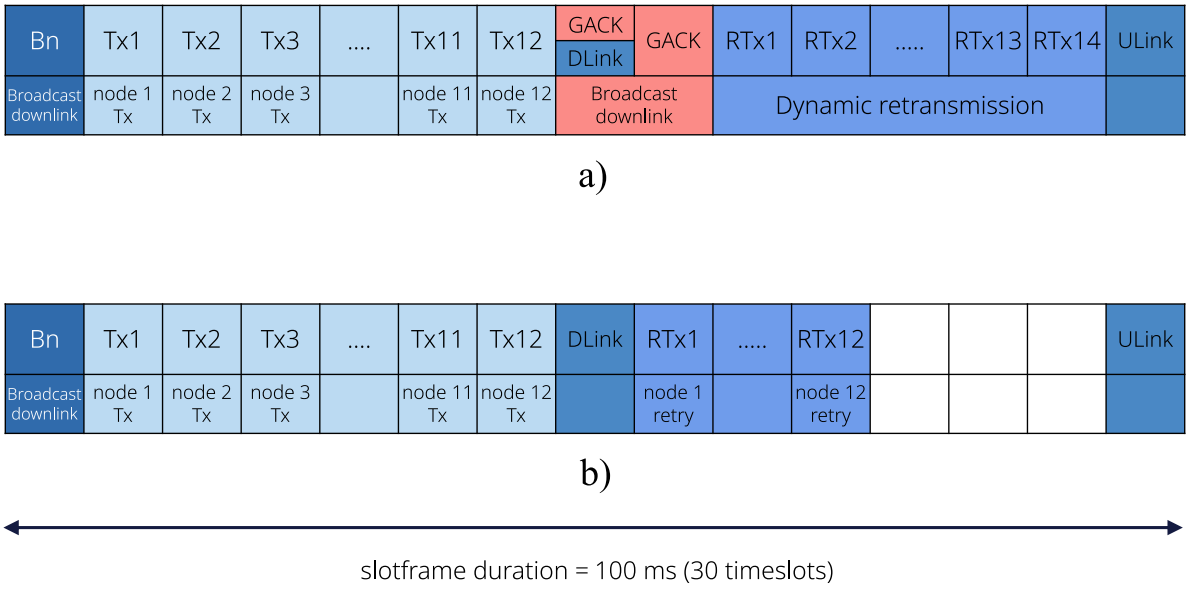


Figure 3.9 – a) Renault Zoe WBMS proposed TSCH schedule which enables high reliability through GACK dynamic retransmission. b) TSCH static retransmission schedule. Both schedules were implemented and compared to demonstrate the advantages of our proposal.

With this schedule definition, we implemented our proposed protocol stack in the hardware described in the previous section. The objective was to create a wireless network that can fulfill all the requirements of the WBMS. In the next section, we present some tests we executed to check if our proposal met the stated objective.

3.3 WBMS network performance

To validate our proposed framework, we executed several experiments to demonstrate that it can provide highly reliable and low power communication with bounded latency. In order to show the qualities of the dynamic retransmission mechanism, we started with some tests where the blacklisting mechanism was disabled. The first one is a comparison between our proposed dynamic retransmission mechanism for the WBMS and a static method. Then, we set the network in a Renault Zoe battery pack to conduct 1-week experiments and present the network statistics collected. Next, we use the TI Energy Trace tool to measure the instant current consumption of the nodes during one slotframe to demonstrate the low energy consumption of our system. Finally, we enable the blacklisting

mechanism and measure the network performance under high external interference.

3.3.1 Dynamic vs static retransmission

Our proposed schedule for critical application uses a GACK-based method to enable dynamic retransmission and achieve high reliability, as presented in section 3.1. In order to show the advantages of this method, we compare its performance against a default static retransmission approach. The contender always uses the same time slot distribution within a slotframe where each node has a dedicated slot to transmit their messages to the root and another to retransmit them in case of failure, as presented in Figure 3.9b.

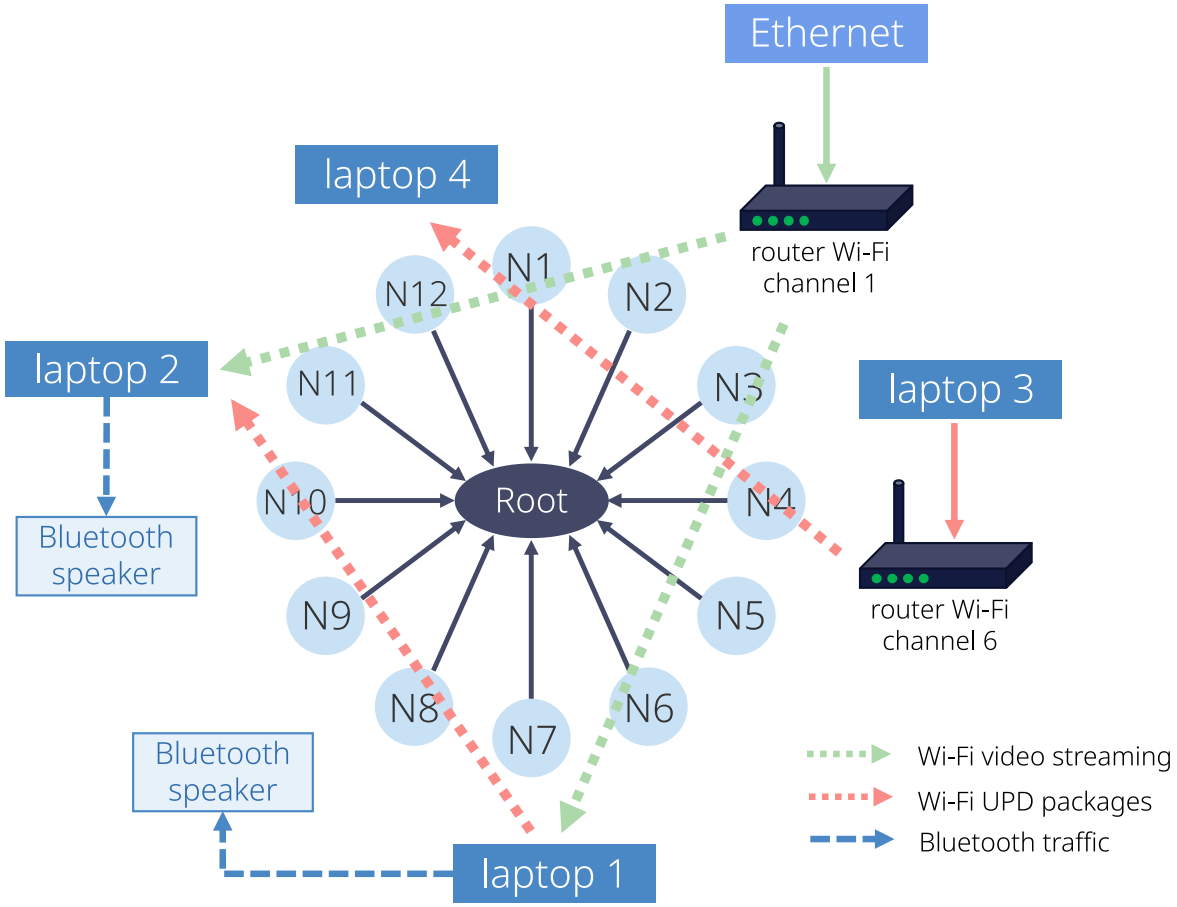


Figure 3.10 – Retransmission methods comparison test setup. For this test we used 13 TI cc26x2 Launchpads for the root and the nodes. Then we use two Wi-Fi networks and two Bluetooth speakers to create electromagnetic interference.

For both cases, we set up a network of 1 root and 12 nodes placed on a desk, using 13

TI Simplelink cc26x2 launchpads (Figure 3.6), where each node sends one frame to the root every 100 ms for two hours. To produce message transmission failures (and retransmissions), we intentionally introduced electromagnetic interference by generating constant traffic over the WiFi radio channels 1 and 6, as presented in Figure 3.10. We placed two routers on the same desk, operating in parallel over two different WiFi networks. Then, we used two computers per network (4 computers in total) to generate constant traffic. Both computers exchanged UDP packets on the radio channel 6 at a rate of 9.6 Mbps using a traffic generator software [1]. For the WiFi radio channel 1, the data rate of the UDP packets was 7.2 Mbps. Additionally, on radio channel 1 both computers streamed videos from the internet. We also put two Bluetooth v4.2 speakers 30 centimeters apart from the nodes, each connected wirelessly to a computer reproducing music during the tests. To collect the data, we programmed the root to send the network statistics over UART to a python script running on one of the laptops. This script stored the data into CSV files which were later processed to obtain the results.

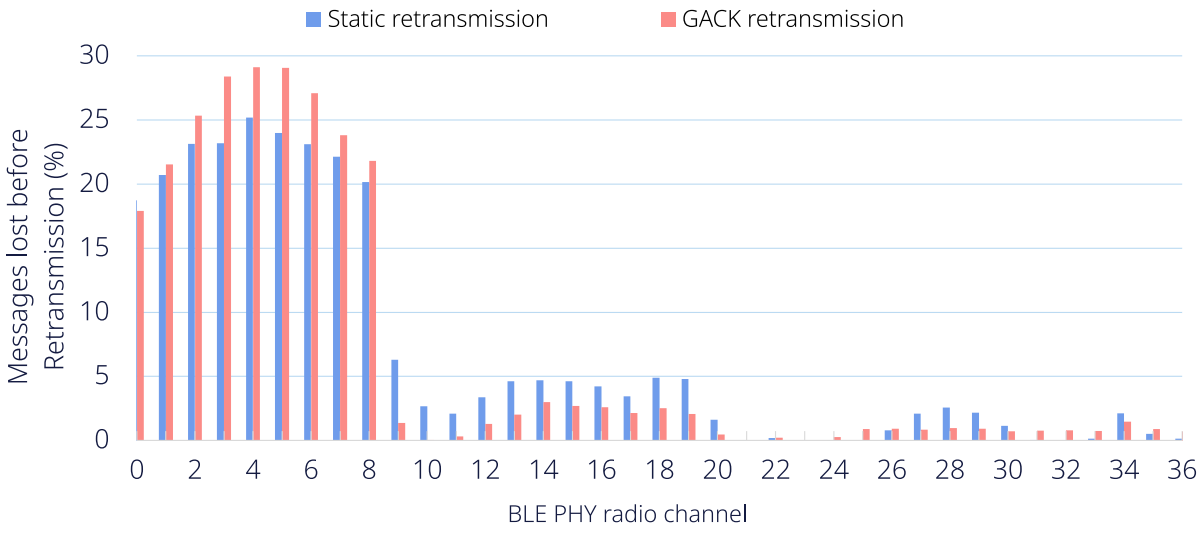


Figure 3.11 – Messages lost per channel before retransmission during both tests. Note that in the two cases the channel stability per channel was similar, so we managed to test the two options under similar external interference conditions.

Figure 3.11 shows the percentage of lost messages per channel before retransmission during the tests. As observed, those BLE channels (from 0 to 9) overlapping channel

1. Packet generator - www.packetsender.com

	Static retransmission	Dynamic retransmission
Total messages	831 957	755 412
Lost before retransmission	58 398	52 218
Lost after retransmission	7277	11
Reliability before retransmission	92.98065 %	93.0874 %
Reliability after retransmission	99.1253 %	99.9985 %

Table 3.2 – Comparison results between our proposed dynamic retransmission method and a static option. The reliability before retransmission is similar, which shows that we generate almost the same level of interference in both cases. Our proposed dynamic GACK-based method outperforms the static algorithm as it lost 600 times less messages.

1 of WiFi suffered high interference. As the generated traffic in the WiFi channel 6 is less than in channel 1, the BLE channels from 10 to 20 were marginally affected while the others (20 to 37) present very few messages lost, which makes sense as there was no intentional interference in this portion of the band. Note that for both cases, the amount of electromagnetic interference was almost equal, so we managed to create similar conditions for both networks. For each case, after two hours the nodes sent almost 800 000 frames to the root, of which around 55 000 were lost on the first try. The network lost 7 277 messages after the second try in the static retransmission case. On the other hand, using our proposed framework with the GACK mechanism, the network lost only 11 messages. Table 3.2 presents the final reliability result for both tests; our proposed schedule reached 99.9985 % of successful message delivery, while the static retransmission protocol was 99.1253 %. So in these particular conditions, our proposal was 600 times more reliable than the static approach.

3.3.2 Renault Zoe WBMS performance

Now that we have shown that our proposed schedule outperforms a standard retransmission protocol, we conducted a series of real world experiments in a Renault Zoe battery pack. The first one demonstrates the network's high reliability and bounded latency capability. The second one presents the low current consumption of the devices over time.

High reliability and bounded latency

In this experiment, we have one root (TI cc26x2 launchpad) and 12 WBMS slaves,

	Without battery cover	With battery cover
Total messages	60 086 640	58 870 164
Lost before retransmission	200 200	21 193
Lost after dynamic retransmission	0	0
Reliability before retransmission	99.6668 %	99.9640 %
Reliability after dynamic retransmission	100 %	100 %

Table 3.3 – WBMS network reliability results after one week tests. Note that the battery cover act as a shield against external interference. The results shows that our proposed WBMS network is able to achieve 100 % of package delivery ratio within 100 ms.

each connected to a battery module. Figure 3.13 shows the setup for this test where the root sends the data statistics to a laptop over UART. During the tests, each WBMS slave measures the eight-cell voltages of a module, groups them in a frame, and transmits it to the root every 100 ms. It is considered a lost message if a frame does not reach the root within a slotframe. The test was run twice for one week each time; the first was without the metal battery pack cover, while the second was with the cover in place.

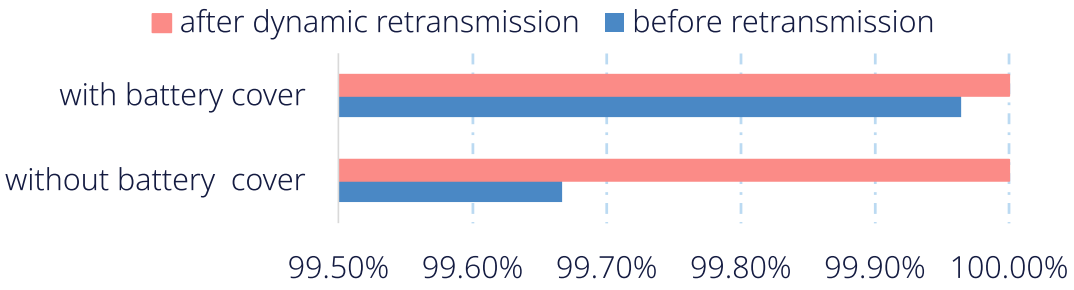
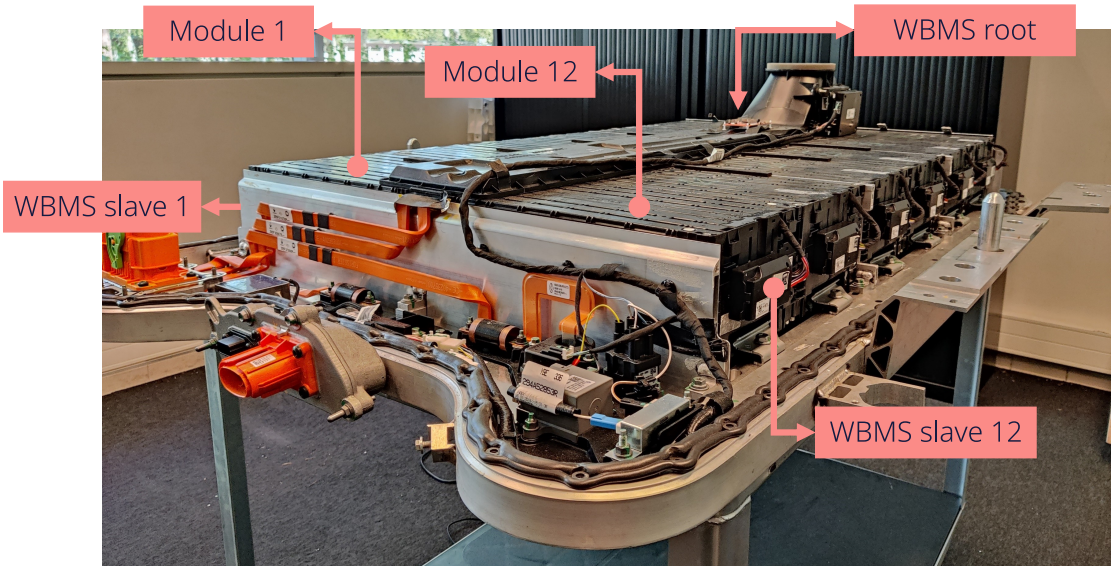


Figure 3.12 – Reliability results of the WBMS network on the Renault Zoe battery pack with and without the metallic cover.

Table 3.3 shows the results obtained from the experiments. For both cases, the total messages exchanged were almost 60 million. By employing our proposed dynamic retransmission schedule, not a single packet was lost, which can be translated as network reliability of 100 %. This shows the outstanding performance of TSCH along with the GACK mechanism and the BLE physical layer. Figure 3.12 illustrates this reliability comparison with and without retransmission for both tests. As observed, the number of lost messages before retransmission is less when the metal cover is in place. It is because the battery pack cover acts as a shield against electromagnetic interference, which is an environmental advantage for the wireless network in this particular application.



a)

b)

Figure 3.13 – a) Reliability test setup in a Renault Zoe battery pack without its cover. The WBMS master is in the center of the pack and each slave is connected to a module. b) Reliability test in a Renault Zoe battery pack with its cover. The root and the slaves are placed inside the cover.

	Mean current (mA)	Mean power (mW)
WBMS root	2.47	8.17
WBMS Node 4	0.42	1.39

Table 3.4 – WBMS nodes average power consumption measured with TI Energy Trace using the TI cc26x2 Launchpad.

Low power consumption

To analyze the power consumption of the root and the nodes, we set up our WBMS network with 11 WBMS slaves, 1 TI cc26x2 Launchpad slave, and the root. Then, we measure the current consumption of the root and the node using the TI EnergyTrace Technology as described in [112]. Figure 3.14 shows the current consumption profile of both devices during a slotframe (100ms). The graph allows us to identify the power modes of the cc2652 microcontroller. The regular operation of the microcontroller is in active mode. When it has finished all its tasks, the device can go to the idle mode or standby mode only if it has enough time to wake up before the next task. This is how we identify the timeslot operation during the slotframe. In figure 3.14, the device is in active mode when the current consumption is higher than 2 mA. When there are current peaks above 9mA, the radio is turned on.

As observed, the root is active between timeslot 2 and 15, receiving frames from 12 nodes and sending the GACK twice. As indicated by our proposed schedule, it is active during the first timeslot and the last (beacon and ULink). Regarding the node, whose ID is 4, it is active in the first timeslot, then in the fifth one to send its frame, and finally during both GACK timeslots. When the devices finish the timeslot operation, they go to the idle mode, where the current consumption is around 1.2 mA. If there is enough time before the next task, they go straight to standby mode, with ultra-low-power consumption, i.e., almost 0. Table 3.4 shows both devices' measured average current and power consumption; as expected, the root consumes more energy than the slave. If a 2.2 Ah 2xAA battery pack powered the slave, the network would run for 218 days, having a constant refresh rate of 10Hz for all the nodes. In the case of a Renault Zoe battery module, whose capacity is 148 Ah, the network would run for 42 years approximately. With these results, we can affirm that replacing the wired link with a wireless network in the BMS would not produce a noticeable negative impact on the EV autonomy.

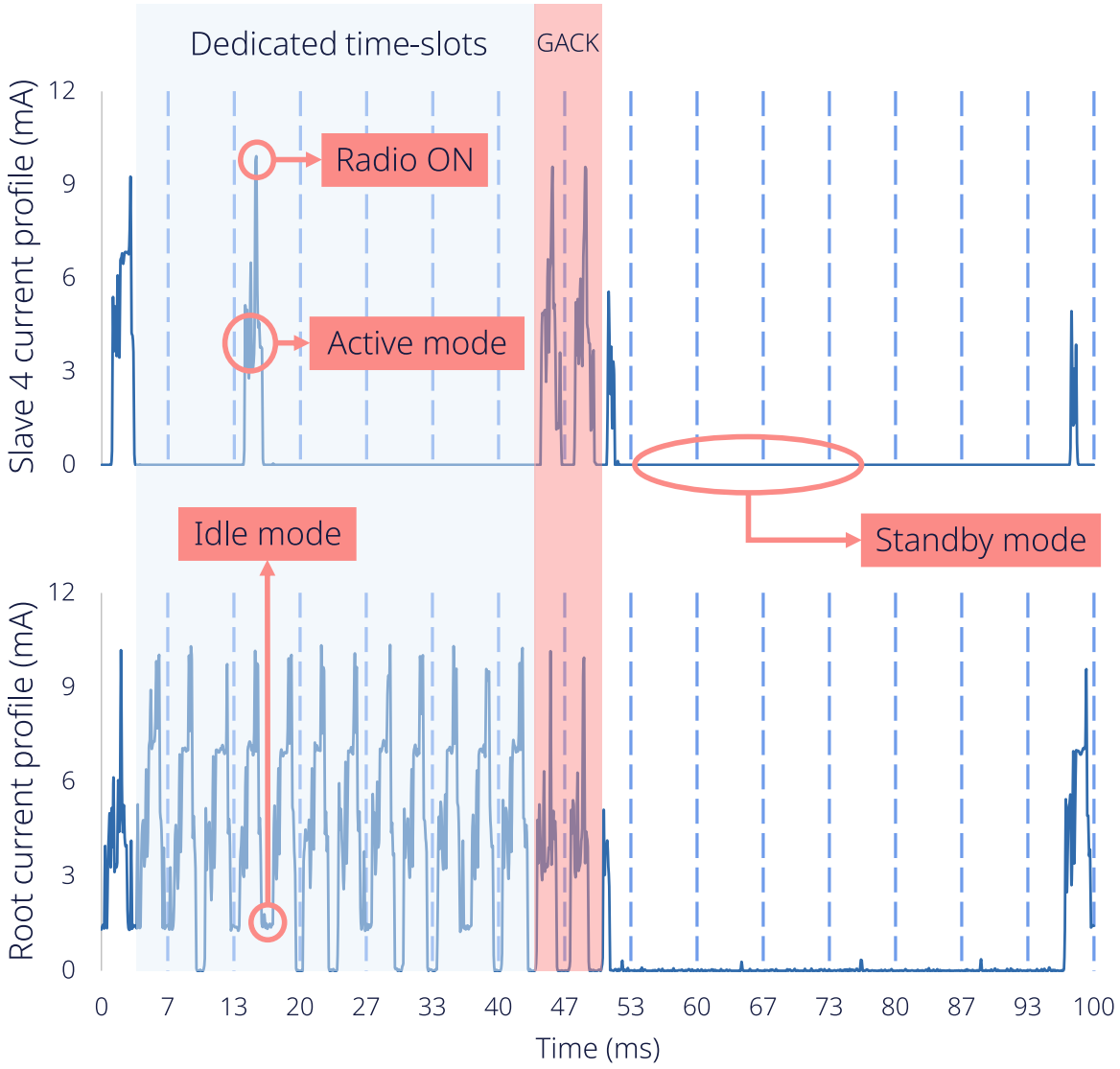


Figure 3.14 – Current consumption over the time during 100 ms of the root and the node 4. The data was obtained using the TI Energy Trace tool and the TI cc26x2 Launchpad. This profile allow us to identify the timeslot operation of our proposed network for high reliable wireless communication. It is also possible to observe the different power modes of the cc2652 microcontroller, when the radio is turned ON it has the highest current consumption, when there is no tasks to execute the node can go to idle mode or standby mode only when there is enough time to wake up before the next task.

3.3.3 Channel blacklisting performance

Until this point, the tests done on our proposed network for the WBMS were executed without the channel blacklisting mechanism because we wanted to show the benefits of using dynamic retransmission management to achieve high reliability, bounded latency, and low energy consumption. However, even a good retransmission policy cannot achieve the required performance in the presence of high external interference. To prove this, we execute a 30-minute test using a similar setup of Section 3.3.1 to generate interference using two Wi-Fi networks. As depicted in Figure 3.15, in this case, the network was deployed in the Renault Zoe battery pack without its metallic cover, and the Wi-Fi networks were set in channels 6 and 11. To generate a significant amount of constant traffic in both networks, we use the Secure Copy Protocol (SCP) to exchange a large file (15 GB) between the computers of the network without any limit of data rate (as presented in [105]).

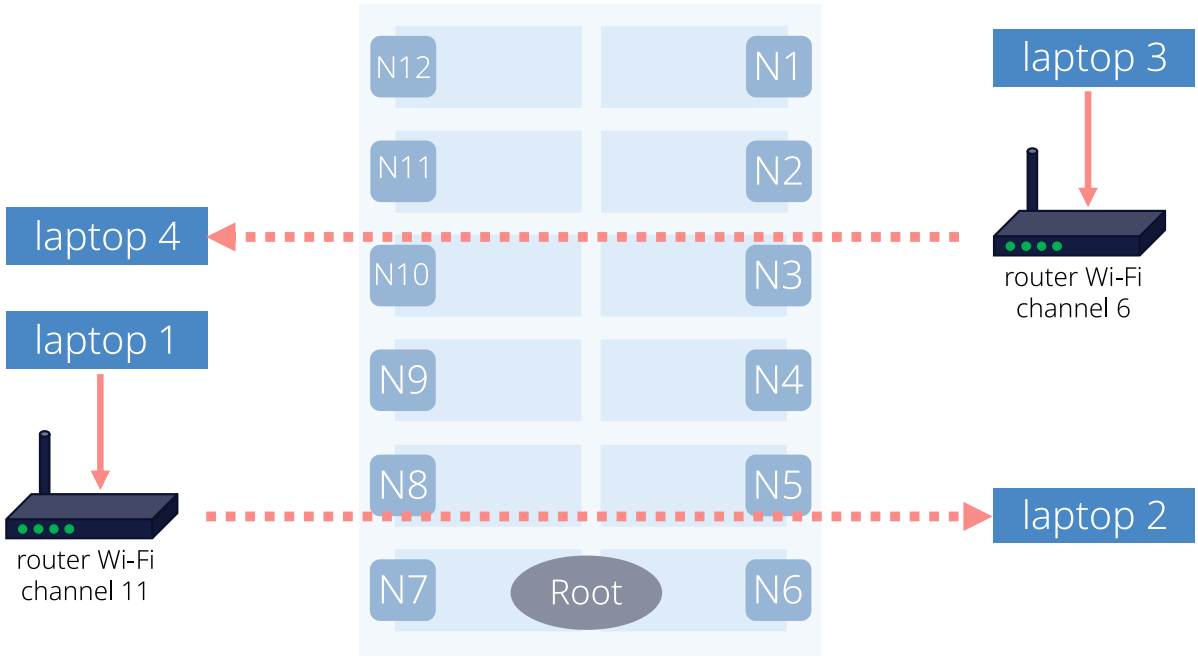


Figure 3.15 – Channel blacklisting performance test setup. The network was deployed in the Renault Zoe battery pack with each slave connected to one module and the root placed between nodes 6 and 7 in the same place of the BMS. The Wi-Fi routers and the computers were placed 1 meter away from the battery pack. In order to achieve the maximum interference we did not put the metallic battery cover in place.

In our test, the interference was enabled right after we started to collect data from

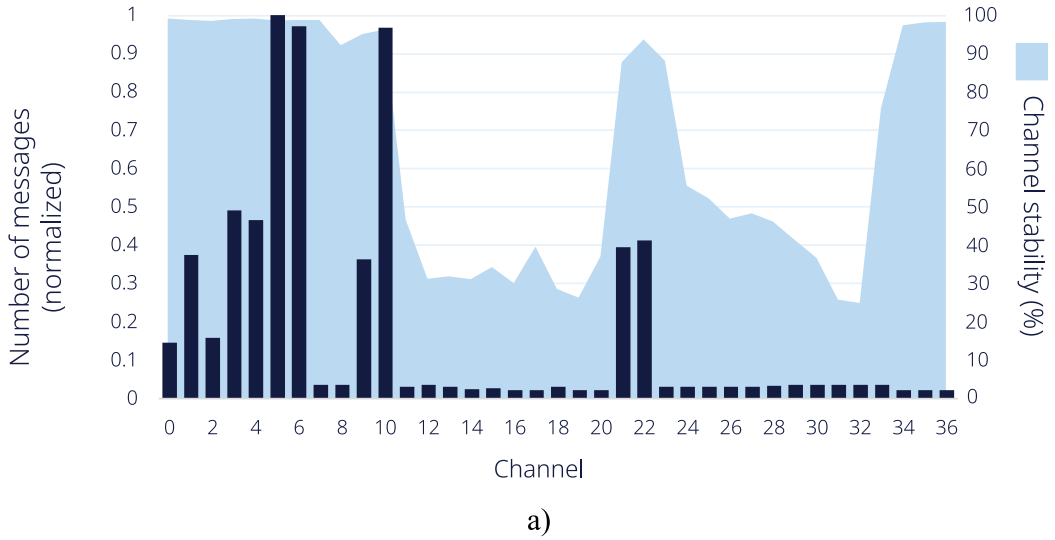
	No blacklisting	**Blacklisting** $\alpha = 0.3$	**Blacklisting** $\alpha = 0.7$
Total messages	216 000	216 000	216 000
Lost before retransmission	74 667	12 634	9 260
Lost after retransmission	15 574	13	26
Reliability before retransmission	65.431%	94.15%	95.712%
Reliability after retransmission	92.789%	99.993%	99.987%

Table 3.5 – 30-minute tests results with and without channel blacklisting comparison.

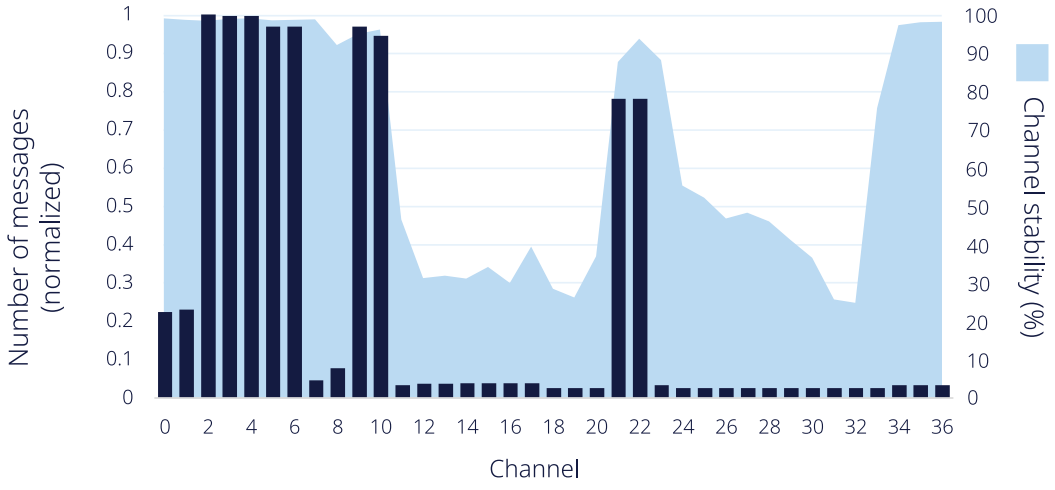
the node; in this way, we can observe how the interference affects the network over time. After 30 minutes, the root is intended to receive 216 000 messages, from which it lost 74 667 before retransmission (dedicated slots) and 15 574 after the retransmission. As you can observe, the generated interference lowered the reliability before retransmission to 65.43 %, and the dynamic retransmission mechanism only reached 92.78 % of final reliability, which is very different from the results obtained in the previous sections. This low performance can be explained by the stability of the channels (Equation 3.2) presented in Figures 3.16a and 3.16b (light blue area). You can notice that 21 channels presented stability lower than 60 %, with some of them under 30 %. As the network continues to use the default hopping sequence with all the channels, when there is a packet lost in the dedicated timeslots, the retransmissions may use some other busy channels. In the worst scenario, the GACK packets may also use these bad channels, which would cause not to have retransmission at all.

$$Channel\ stability(i) = \frac{Number\ of\ messages\ successfully\ delivered\ (i)}{Total\ number\ of\ messages(i)} * 100\ \% \quad (3.2)$$

Section 3.1.3 presented our proposed channel blacklisting algorithm for the GACK-based scheduling function. This algorithm is based on some alternatives reviewed in Section 2.2.1, with additional features to keep the GACK high reliability and manage the blacklisted channels' reintegration. This algorithm was implemented for the WBMS network and tested under the same rough conditions of the previous experiment. For this particular application, we set to 5 the minimum number of good channels, the update time was 30 seconds, beta was set to 0.01, and we tested two values of α: 0.3 and 0.7. For the blacklist time, we set 2 minutes for the first level, 10 minutes for the second, and



a)



b)

Figure 3.16 – Number of messages per channel during the black listing tests normalized using the channel with the highest number of messages. a) $\alpha = 0.3$. b) $\alpha = 0.7$. In both cases the light blue area represent the channel stability obtained from the test without the blacklisting enabled to illustrate the interference to which the network was subjected.

20 for the third level. After 30 minutes, the root lost 12 634 messages with $\alpha = 0.3$ and 9 260 with $\alpha = 0.7$ before retransmission, which represents an 83 % reduction with respect to the default channel sequence. The results show that after retransmission, using $\alpha = 0.3$, **the root lost 13 messages in the first 39 seconds, and then it did not lose any more during the rest of the test**. Using $\alpha = 0.7$, the root lost 26 messages after retransmissions in the first 30 seconds, with 0 messages lost after that. Table 3.5 compares the results of the three tests described before.

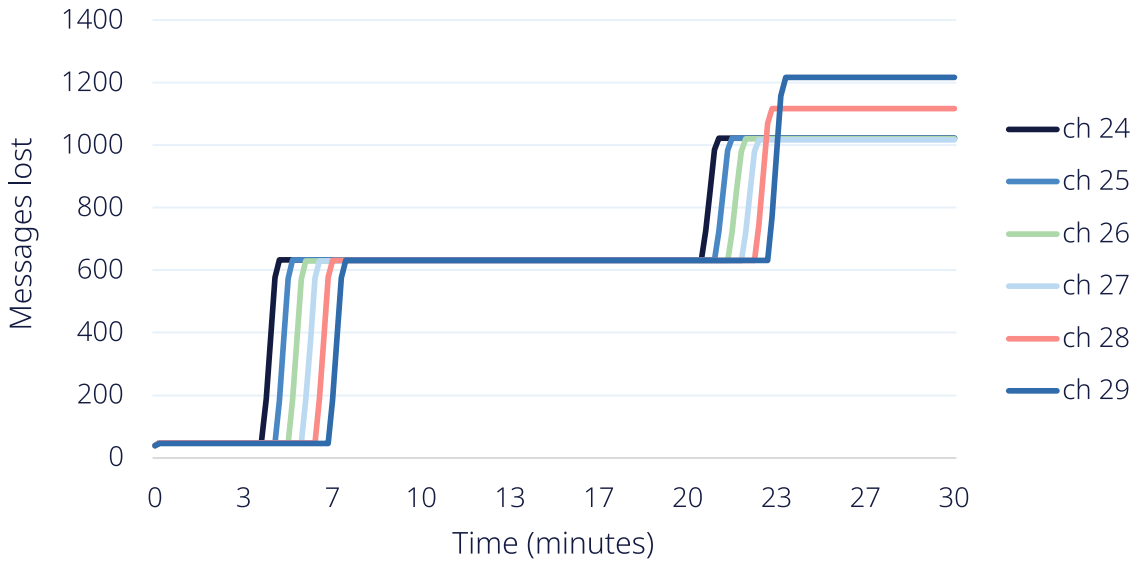


Figure 3.17 – Number of messages lost over time using a set of channels with high interference, as observed in Figure 3.16. This plot illustrates our proposed channel reintegration mechanism which uses different ban durations depending on the level of the channel and test only one blacklisted channel at the same time.

We have seen that the algorithm is tunable, as several parameters can be changed depending on the application. Figure 3.16a and 3.16b present the number of messages transmitted per channel for the test with $\alpha = 0.3$ and $\alpha = 0.7$, respectively. The results are normalized using the channel with the maximum number of messages transmitted during the 30-second tests. To make it clearer, we added the channel stability results obtained from the first test (light blue area). As you can observe, when using $\alpha = 0.3$, the utilization of the good channels was uneven. This exemplifies how a small weight factor downplays the previous state, and the channel blacklist changes considerably at each refresh period. On the other hand, with $\alpha = 0.7$, we obtained a better use of the good channels as several have a similar number of messages transmitted. So, the channel

blacklist does not change between the refresh period as much as in the previous case. To illustrate the behavior of the reintegration algorithm, Figure 3.17 shows the number of messages transmitted over time using a set of channels that presented low stability. Once the channels are blacklisted at the beginning of the test, they are tested after a while (2 minutes minimum) for 30 seconds and blacklisted if the interference is still present. Please observe that the second interval is longer than the first one, representing the different levels proposed by our algorithm.

The results presented in this section show that our proposed blacklisting method for the GACK scheduling function reduces the number of messages lost and the number of retransmissions needed to achieve high reliability. Also, our rule of not using the GACK or retransmission timeslots for testing the reintegration channels produces very good results for the final performance. We have seen that the messages lost after retransmission occurred at the beginning; this is because the algorithm needs T_{update}, 30 seconds in our case, to identify the changes in the stability of the channels.

3.4 Conclusion

Critical applications require communication systems with high reliability because failures may produce catastrophic consequences. Typically, wired links are the preferred option, even if a wireless option is easier (cheaper) to install and maintain. In this chapter, we proposed a wireless protocol framework for critical applications which enable high-reliable wireless communication with deterministic latency and low power consumption. Our proposal uses the IEEE Std 802.15.4 TSCH MAC layer, which uses time division and frequency diversity to be robust against external interference. On top of this layer, we design a scheduling function using some features of LLDN, such as the GACK mechanism, to enable dynamic retransmission management of the packets within a slotframe. This method adapts the allocation of the retransmission time slots from one slotframe to another according to the messages lost in the dedicated links. We consider that the physical layer of the IEEE Std. 802.15.4 is not sufficient to meet the latency requirements of some critical applications. Therefore, we proposed to employ the BLE PHY 2Mbps layer in conjunction with a new timeslot structure to achieve shorter timeslot duration and higher refresh rates. Our proposal also includes a channel blacklisting algorithm specially adapted to our scheduling function in order to add robustness in the presence of high external interference.

Once we have explained the operation principles of our network, we use this proposal to design and implement the active mode of a WBMS for the Renault Zoe battery pack. We first discuss the system requirements; refresh rate, energy consumption, and reliability. Then, we present the software and hardware used to implement our solution; we have chosen the TI cc26x2 RF microcontroller family for the network management and the TI BQ79616 BMS ASIC for the voltage measures and cell balancing control. Next, we validate our solution by conducting a series of real-world experiments, showing that our proposal outperforms a traditional static retransmission protocol and can achieve 100% of end-to-end reliability with bounded latency and low energy consumption. Finally, we tested our network under high external interference, where we have shown the importance of the blacklisting mechanism in these kind of rough environment. Our experimental campaign demonstrates that the GACK mechanism, in conjunction with the TSCH MAC mode, allows us to enable wireless communication on critical applications such as the WBMS without compromising the safety level of the overall system.

Now that we were able to design and implement the active mode of a WBMS where the root receives a new measure periodically, the following steps include the definition of the sleep mode to achieve a complete WBMS. Once the system is complete, it is necessary to integrate this new interface with an EV and execute a series of driving experiments to evaluate the wireless network in multiple scenarios.

Chapter 4

Sleep mode for WBMS

The BMS has two modes of operation, active and sleep mode. When the EV is driving or charging, the BMS periodically collects the battery cells data (e.g., every 100 ms), as described in the previous chapter. When the vehicle is parked (without being connected to the charger), the BMS goes to sleep mode, where it only has to wake up (e.g., every 8 hours) to check the charge balance of the battery cells. A BMS should be able to switch from sleep to active mode within a specific time (1 second for the Renault Zoe). Also, the energy consumption in this mode should be negligible, i.e., less than $70\,\mu A$ per slave for the Renault Zoe. In a wired BMS, turning off the slaves during sleep mode is easy because we can wake them up through an interruption sent over the UART bus. For a WBMS using a synchronized network, we cannot just turn off the slaves entirely because we do not have any physical link to send a wake-up interruption. In fact, it is necessary to maintain the nodes synchronized; otherwise, the slaves must keep the radio on during the sleep mode listening to the medium until they receive the wake-up order, which would consume so much energy (8 mA on average for the CC26x2 microcontroller).

The easiest way to implement the sleep mode for a WBMS is to create a new schedule where the master periodically broadcasts a beacon (EB) message to keep the network synchronization, as depicted in Figure 4.1. In this case, the wake-up process is as simple as sending the command in an EB; when the nodes receive it, they switch to active mode. The interval between EB messages depends on the wake-up time required by the application; for the Renault Zoe, we may use 1 second. However, if one node does not receive the wake-up command, the whole system must wait for another second to change to active mode, so it would be better to use a smaller value (e.g., 400 ms). Note that there is a tradeoff between energy consumption and wake-up time. The smaller the wake-up time requirement, the higher the power consumption.

This WBMS sleep mode has an essential issue. Keeping the WBMS master awake is necessary to maintain the network synchronized. Unlike the slaves, which are powered directly by the battery modules, the master is powered by a small 12 V battery with

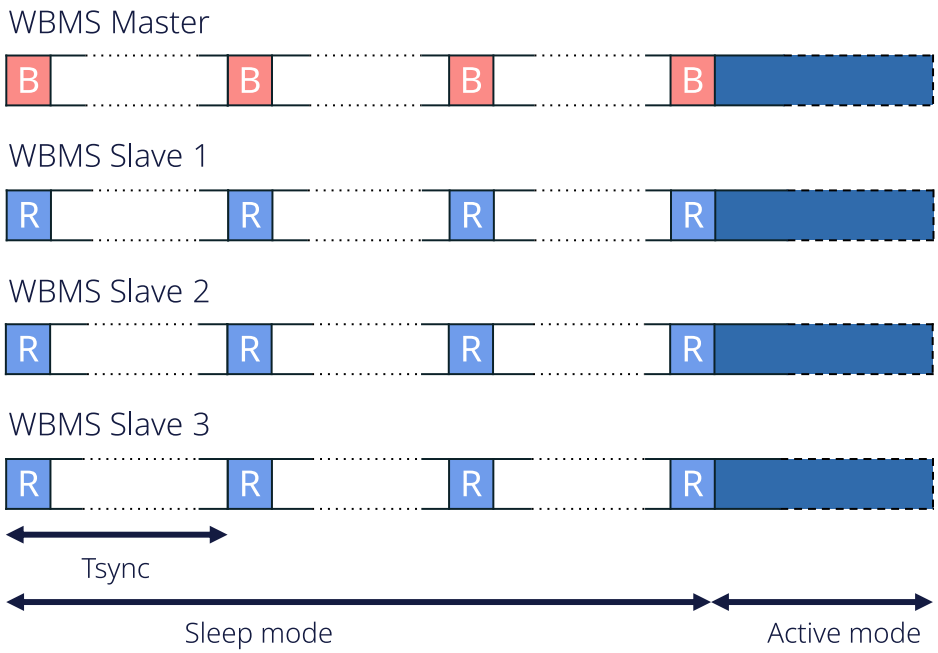


Figure 4.1 – Typical approach for a low-power mode on synchronized wireless networks.

much less capacity than the battery pack. Recharging this 12 V battery from the battery pack is possible, but this represents a waste of energy because DC-DC conversion does not have 100% efficiency, and it is necessary to wake up some other components of the vehicle. Also, when the battery SOC is below a threshold, the BMS does not allow this recharging process. Therefore, in this chapter, we propose an alternative to keep synchronization between the nodes without using the master in sleep mode—all of this without exceeding the wake-up time and the energy consumption limits.

4.1 Synchronization without the root

In this section, we present our sleep mode for the WBMS network, where it is possible to turn off the WBMS master and keep the battery pack balanced. First, we present the basic principle to keep the synchronization without the central device, then we detail the transitions from active to our proposed sleep mode and vice versa. Last, we present the energy consumption and wake-up time results of this new mode applied to our WBMS network implementation for the Renault Zoe battery pack.

Our WBMS network has a star topology, where the slaves synchronize their clocks

when they receive a message from the master. If we turn off the master in sleep mode, we need another source time for the nodes to keep their clocks synchronized. A straightforward solution to this problem would be to set one slave as the time reference of the network while the master is turned off. In this case, the selected node should send EB messages periodically, receive the battery cell voltage measurements from the nodes, and broadcast this data to keep the battery pack balanced. This behavior would produce this node to consume more energy, thus contributing to increasing the battery pack unbalance.

To avoid this problem, we propose not to choose a unique central node to synchronize the network but to use all the nodes with a pre-defined schedule. To keep synchronization, every `Tsync` seconds, one different node must broadcast an EB to reduce the clock drift between the slaves. This means that every node turns on its radio every `Tsync` to receive an EB from another node or to send an EB when its turn comes. With this method, every node has the same workload, and consequently, all have the same energy consumption. Additionally, each node can include the voltage measurements of its battery module in the EB message. In this way, every node receives all the battery pack cell voltages and can decide which cells of its module need to be discharged to keep the battery pack balanced.

To change from sleep to active mode, the WBMS master must resynchronize to the network and be the time source of the slaves again. When the master receives the wake-up command from the EV central computer through an interruption, it should turn on its radio and listen for an EB of a WBMS slave. As soon as it receives the synchronization message, it can start sending EBs, telling the slaves to change to active mode. In this new sleep mode schedule, each WBMS slave wakes up every `Tsync` during two timeslots. In the first timeslot, the nodes send or receive an EB from another slave as described before. In the second one, the nodes listen to the radio channel and wait for an EB of the master.

Figure 4.2 illustrates the proposed sleep mode with an example of a network with four nodes and one master. As you can observe, in the first part, the master is turned off, and the slaves wake up every `Tsync` during two timeslots. In the first timeslot, a different node sends a beacon (white B) while the others listen to the radio channel to receive the message and synchronize their clocks (white R). In the second timeslot, the nodes are listening for a beacon from the master (black R). When the master receives the wake-up interruption, it turns on the radio and listens to the channel for a beacon from a node (white R). When it receives the EB from node 4, it can synchronize its clock, send an EB (black B) to wake up the network, and go to active mode. It is important to remember that the wake-up time length depends on the `Tsync` value and the moment the root

wants to rejoin the network. If the master receives the wake-up interruption just after a node has sent the EB, it would have to wait `Tsync` until the next EB of the network to synchronize its clock and send its beacon, as depicted in Figure 4.2. Conversely, in Figure 4.3, you can observe that if the master receives the interruption just before the EB from a node, it will synchronize almost instantly.

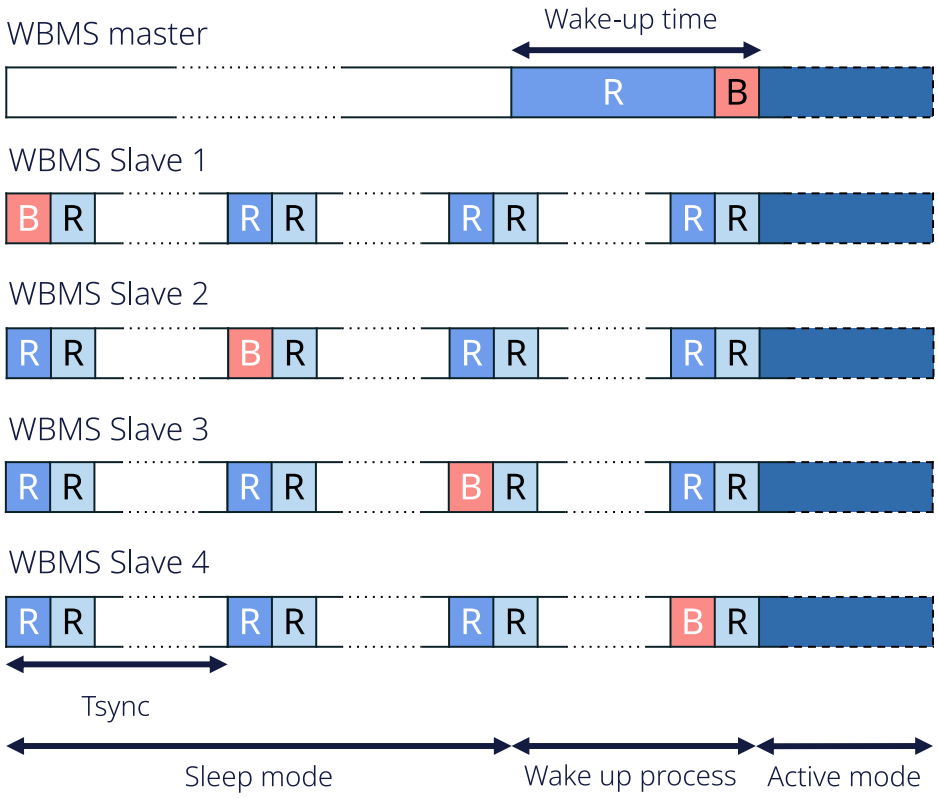


Figure 4.2 – The proposed sleep mode schedule for WBMS networks (left part of the timeline), every Tsync a different node send a beacon to synchronize the network. The right part of the timeline shows the worst case scenario for the wake-up sequence which results in long wake-up time.

In our particular case, for the Renault Zoe WBMS, where the maximum wake-up time (WuTimeMax) is 1 second, it is necessary to choose a `Tsync` smaller than WuTimeMax. However, because the network must ensure that all the nodes have received the wake-up order before one second, the master should be able to send at least two EB to the nodes within this interval. For this reason, we propose to use the relation described in Equation 4.1. Note that the energy consumption of the slaves in sleep mode also depends on the synchronization period; if `Tsync` is too small, the slaves would consume more energy.

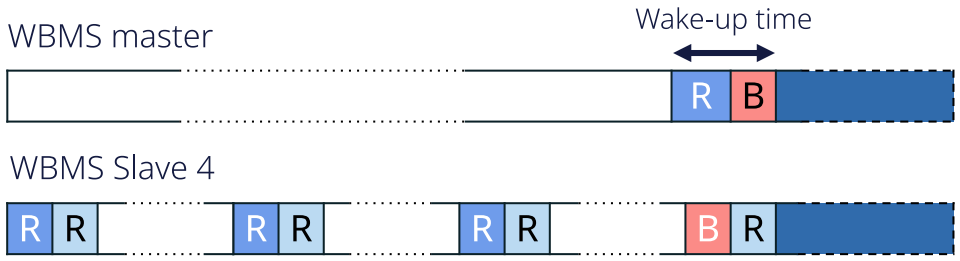


Figure 4.3 – Best case scenario for the wake-up sequence which results in a short wake-up time.

$$Tsync < 0.5 * WuTimeMax \tag{4.1}$$

4.1.1 Renault Zoe WBMS sleep mode

The proposed WBMS sleep mode was implemented for the Renault Zoe battery. We use the same hardware and software presented in the previous section with the custom board for the WBMS slaves composed by a radio microcontroller (TI cc2642) and a BMS ASIC. For the master we used a TI cc2652 Launchpad.

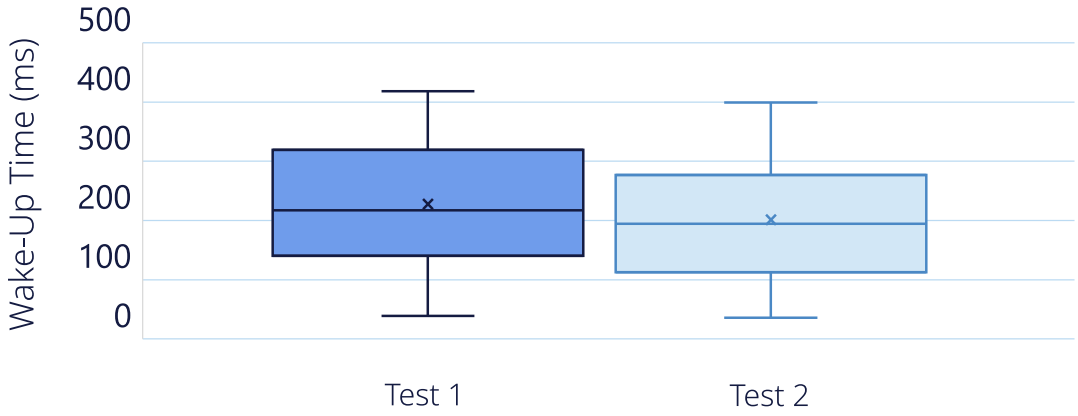


Figure 4.4 – Wake-up time experiment results. Using $Tsync = 400\ ms$ we observe a maximum wake-up time of 418 ms and a minimum of 36 ms among 200 repetitions.

To achieve the required wake-up time we chose $Tsync = 400\ ms$ following Eq. 4.1. The network was tested in sleep mode for 3 days, and the nodes never lost synchronization. We also measured the energy consumption of the slaves by using the TI Energy Trace tool,

the average current consumption obtained was 60 μA per slave. To measure the wake-up time, we conduct two tests where the master was configured to set up 100 times the network in our proposed sleep mode, during a random time between 8 and 12 seconds, and go back to active mode. We collected the wake-up time value for each cycle, and present the results in Figure 4.4 for both experiments. As you can observe, the network never took more than 450 ms to pass from sleep to active mode, and in most of the cases the wake-up time was between 100 ms and 300 ms. These results show that with our proposed sleep mode for the WBMS network, it is possible to keep synchronization while the master is turned off, and, at the same time, we can achieve ultra-low power consumption for the nodes and a wake-up time below the required limit.

4.1.2 Ultra-low power sleep mode

In this section, we discuss how we can further reduce the energy consumption of the WBMS slaves in sleep mode. The previous section presents our sleep mode where the nodes wake-up every `Tsync` to synchronize its clock, and to listen for an EB of the master. To maintain the wake-up time below the maximum limit imposed by the application, the nodes must keep sending EB every `Tsync` (white B slots in Figure 4.2), so the master can rejoin the network once it receives the wake-up interruption. At the same time, the slaves must listen for an EB from the master also every `Tsync` to receive the wake-up order as soon as the master has been resynchronized to the network (black R in Figure 4.2). As we mentioned in section 2.2.1, in a TSCH network the nodes must resynchronize its clock at most every `Tmax`, which is defined by Equation 2.6. Nevertheless it is recommended to use `Tmax/2` as the maximum period so if a node misses a beacon message, it has the opportunity to receive another one before losing synchronization with the network. This means that the nodes do not necessarily have to resynchronize their clocks every `Tsync`, but every `Tmax/2`.

In our case, the TI cc26x2 has a clock drift of ± 40 ppm, so in the worst case, two microcontrollers would have a 80 ppm clock drift, which is 80 μs every second. The WBMS implementation that we are using, which is described in the previous chapter uses $RxWait = 300\mu s$. According to Eq. 2.6, the maximum period of resynchronization for the nodes in this particular case is:

$$Tmax\ (s) = \frac{RxWait \quad (us)}{ClkDriftSec \quad (us)} = \frac{300}{80}\ (s) = 3.75\ (s)$$

Therefore, in this network the nodes can synchronize their clocks every:

$$\frac{Tmax}{2}\ (s) = \frac{3.75}{2}\ (s) = 1.87\ (s)$$

Which is almost 4 times the `Tsync` chosen in the previous section. According to this analysis, to decrease the energy consumption of the WBMS slaves in sleep mode, it is possible to reduce the number of timeslots per second that each node use to resynchronize its clock (white R in Figure 4.2). However, it is important to choose the appropriate timeslots, so we do not produce "synchronization loops" or "synchronization swing" problems as described in Section 2.2.1.

Figure 4.5a represents the sleep mode proposed in the previous section for a network of six nodes. As you can observe, every Tsync, one node sends a beacon message (white B), and the others receive it and synchronize their clocks (white R). A first approach to reduce the number of timeslots where the nodes listen to the radio channel (white R) may be to not wake up every Tsync after sending their beacons but to do it every $2 * Tsync$, as depicted in Figure 4.5b. Theoretically, it should work because $2 * Tsync$ is smaller than $Tmax/2$. If we look closely at the schedule, it is easy to notice that we have created two synchronization loops because node one always synchronizes nodes 3 and 5, while it is always synchronized by the beacons of nodes 3 or 5. A similar relation occurs for nodes 2, 4, and 6. This schedule was implemented in our software to probe the analysis; after some time in sleep mode, the root could not join the complete network. It was sometimes able to synchronize with nodes 1, 3, and 5 or with nodes 2, 4, and 6, but never with all of them.

Another example is presented in Figure 4.5c; in this approach, the nodes synchronize their clock with the beacon sent by the precedent node. Then, it sends its beacon, which the next node uses to resynchronize its timer. This schedule presents a clear example of the synchronization swing phenomenon presented in section 2.2.1. It may happen that when we synchronize node 5 with the beacon of node 4, which has been updated by the precedent nodes, the drift between nodes 5 and 6 would increase over the guard time value which will produce a loss of synchronization of the node 6. This example was also implemented in our software. After some time in sleep mode, the nodes started losing synchronization progressively until all the nodes were out of the network.

If the network synchronizes all the nodes with the same EB, as is depicted in Figure 4.6, we can avoid synchronization loops and swing problems. For the WBMS of the Renault Zoe, we decided to synchronize the slaves with the EB of nodes 1, 5, and 9. As you can

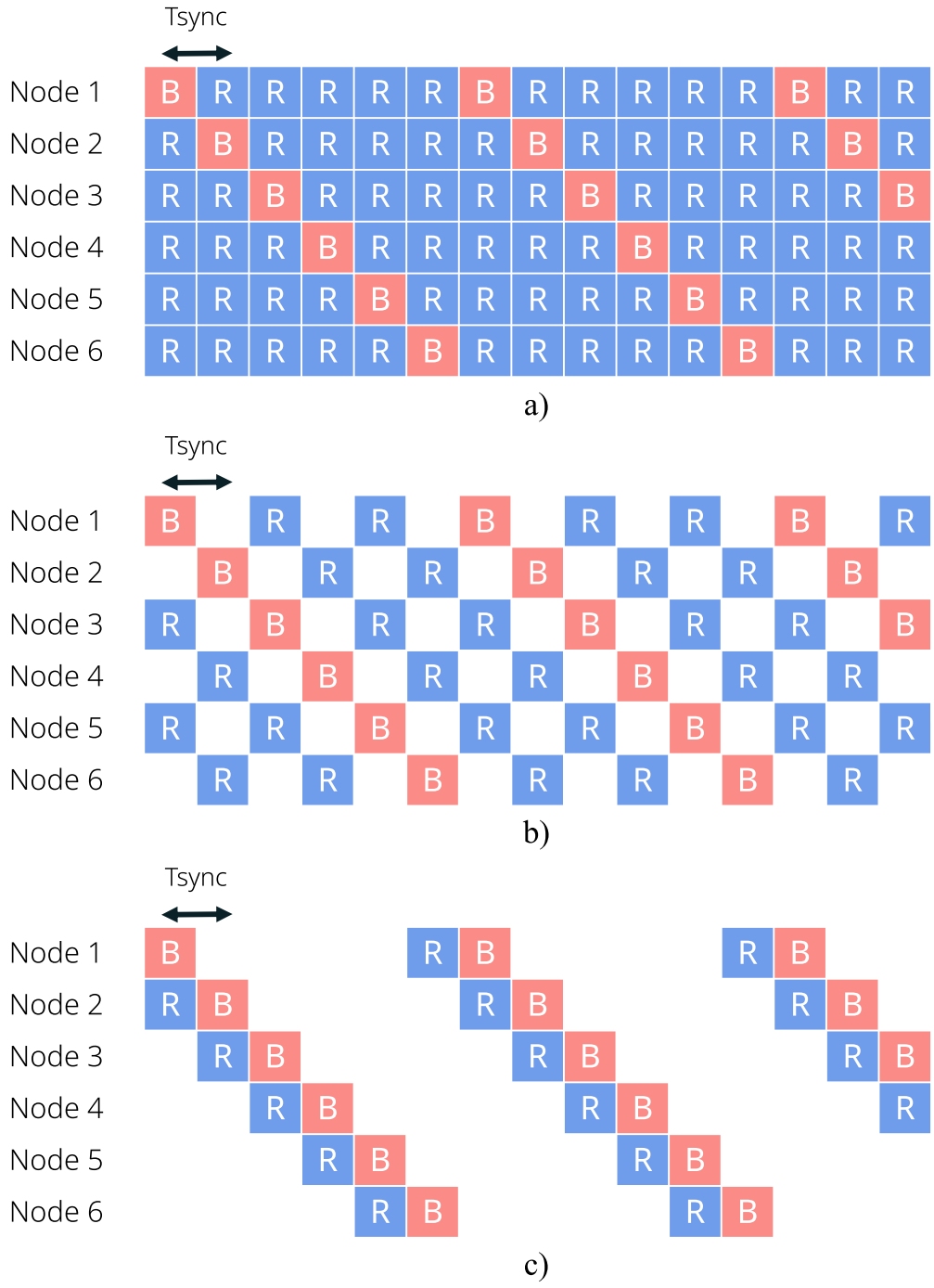


Figure 4.5 – a) Sleep mode schedule presented in the previous section for a network of 6 nodes. All the nodes wake-up every Tsync to receive or send a beacon and synchronize the network. b) Energy reduction attempt for the proposed sleep mode. In this case we create two synchronization loops, so node 1, 3 and 5 will lose synchronization with nodes 2, 4, and 6. c) Another energy reduction example which produces the synchronization swing phenomenon.

observe in Figure 4.6, the network sends an EB every `Tsync`, so the wake-up time remains the same, and they listen for a new EB at the same timeslot every `4*Tsync`, which is smaller than $Tmax/2$. We also added a few extra timeslots to listen for an EB to nodes 1, 5, and 9 to keep the same energy consumption for all the nodes. This method was tested for three days on the same WBMS network of the previous section, and the results showed a 20% reduction in the power consumption without losing synchronization. The measured average current consumption of each slave in sleep mode obtained with the TI Energy Trace tool was $40\mu A$. As you can observe, in this ultra-low power sleep mode, the nodes cannot broadcast their cell voltage measurements to keep the battery pack balanced because not all the nodes listen to their beacons. To solve this issue, the network may use both sleep modes according to the balance of the battery; if the cell balancing is necessary, they use the proposed sleep mode of the previous section. Otherwise, it changes to this ultra-low power mode.

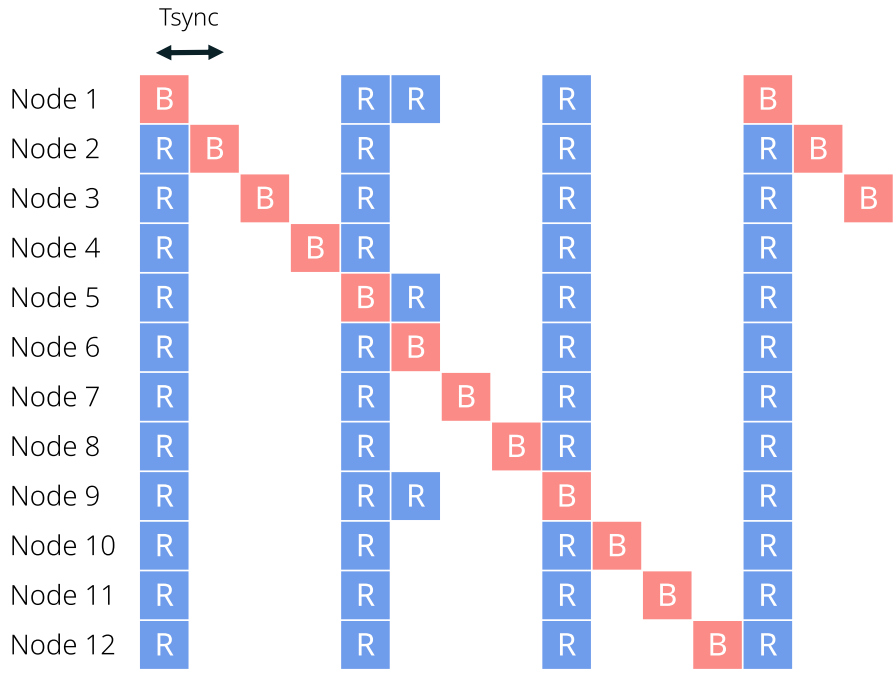


Figure 4.6 – The enhanced proposed sleep mode schedule for WBMS which produces lower energy consumption per slave node keeping the same wake-up time as the previous version.

	Sleep mode	**Ultra low power sleep mode**
Tsync	400 ms	
Max wake up time	418 ms	
Slave average current consumption	60 μA	40 μA

Table 4.1 – Energy consumption comparison between the two sleep modes proposed. The enhanced sleep mode consumes 33.3 % less power than the first approach.

4.2 Conclusion

In a wired BMS, the master must keep the battery pack balanced; to do this, it wakes up periodically (e.g., every 8 hours) to check the battery cell's charge level. This behavior makes the BMS master consume energy from the 12 V battery of the EV, which is an undesirable behavior as it has low capacity, and sometimes it is not possible to recharge in sleep mode. Implementing a wireless link between the master and the slaves in a BMS brings several advantages, but it has multiple challenges too. In this chapter, we have seen that we cannot turn off the WBMS slaves in sleep mode because there is no way to wake them up through an interruption. Also, if the nodes lose synchronization, they should keep their radio on, waiting for the wake-up order, which produces high energy consumption. The most common approach to solve these issues would be to keep the master sending EB with low frequency to synchronize the nodes. However, this needs the master to be active, having the same problem with the 12 V battery as the wired BMS.

In this chapter, we proposed two WBMS sleep modes where it is possible to turn off the WBMS master. The first one keeps the network synchronized and the battery pack balanced. The second only maintains the network synchronization but with ultra-low power consumption. The idea is to alternate both modes according to the balance status of the battery. The principle of operation is that every node wakes up periodically to receive or send an EB. One different node sends the EB each time, and the others synchronize their clock with this message. In this way, all the nodes have the same energy consumption, keep their clock drift low, and share the battery cell voltage measurements to execute the necessary balancing operations. Both modes were implemented on our real hardware platform for the Renault Zoe WBMS. Table 4.1 shows the comparison of the results, the second mode presents 33.3 % less power consumption than the first one, but it does not allow to keep the battery balanced. For both cases, the sleep mode met

the Renault Zoe requirements; wake-up time of less than 1 second and average current consumption below 70 μA. These modes were tested for three days, and the nodes never lost synchronization.

Now that we have a complete WBMS design with the same functionalities as a wired BMS, the next step in our work is to go beyond these essential BMS functions and evaluate which other functionalities a WBMS can perform. Then, we will integrate our WBMS with a Renault Zoe to execute real driving experiments.

Chapter 5

Beyond the wireless BMS

In the previous chapters, we explained what a BMS is and the basic functions it must execute to ensure the safe operation of the battery pack inside an EV. We also presented our network proposal to replace the wired interface between the BMS master and the slaves with a wireless link. We executed several laboratory tests to prove the feasibility of this wireless network introduction in such a critical system. In a traditional BMS, there is only one microprocessor in the system, which is in the BMS master. This microprocessor is not only in charge of performing all the BMS battery calculations (SOC, SOH, cell balancing), but it also must control and watch over the BMS ASIC of each slave. WBMS requires the inclusion of an RF microcontroller in each slave, which has a physical interface with the BMS ASIC and oversees the wireless communication management. Therefore, in a WBMS, we gain a distributed computing capacity compared to a traditional BMS. Even though these microcontrollers do not have super powerful microprocessors, they can still perform some operations to reduce the primary device's workload. Introducing a WBMS is an opportunity to change the BMS central computing operation to enhance some characteristics of the system and reduce costs. In fact, some calculations can be made in parallel, and the central device does not have to be as powerful as in the traditional BMS systems. In chapter 4, we already introduce this topic with autonomous balancing, where the network could keep the battery pack balanced during sleep mode without the intervention of the WBMS master. In this chapter, we propose and test some alternatives of distributed computing for our Renault Zoe WBMS implementation.

In a traditional BMS, when the system needs a new voltage or temperature measurement from all the battery cells, it must send a command to trigger this operation in the BMS ASICs and wait for the response. In some cases, the BMS may need multiple measurements within the refresh time of the algorithms to compute an average and reduce the variance, increasing the estimations' precision. Additionally, these measurements must be synchronized with the current measurement; otherwise, a SOC estimation algorithm (e.g., Kalman filtering) may present inaccuracies in its results. In our WBMS for the Renault

Zoe, the WBMS master receives a new message from the slaves every 100 ms. In this case, we have a software application in the nodes that periodically sends a command to the BMS ASIC to measure the cell's voltage, wait for the response, and transmit the data to the WBMS master. Note that we already use the computing capacity of the slaves to avoid the measure trigger message from the WBMS master. In a wired BMS, sending multiple commands within an interval of 100 ms to take multiple measures and compute the average is easy because the wired link is always available. In a WBMS it is not the case, but we can use the slaves' microprocessor to execute this task and take advantage of the synchronized network to retrieve all the battery cell measures simultaneously.

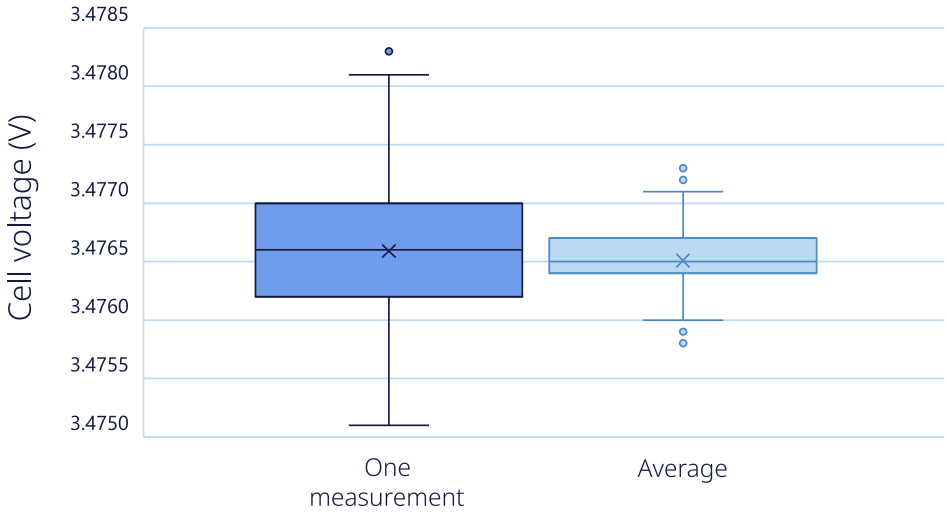


Figure 5.1 – Precision comparison between one voltage measure every 100ms and the average of 5 measures during the same time interval taken every 20ms.

We implemented this solution in our WBMS application synchronized with the time slot-based link layer. First, the nodes wake up to retrieve a new measure from the BMS ASIC in slots 3, 9, 15, 21, and 27 (every 20 ms). Then, each slave computes the average of the five measurements for each of the eight cells in its module. Finally, it transmits this data to the master in the next slotframe. With this solution, the master receives the final average value every 100 ms without sending a new command every 20 ms or performing 96 average calculations. To probe the advantages of this method, we set up our network in a Renault Zoe battery pack to measure the battery cell voltages during 60 seconds with and without the average operation. Figure 5.1 shows a boxplot of the voltage measurements of a cell for each case. As you can observe, without any intervention of the WBMS master, the difference between the maximum and minimum measures was 3 mV

with a single measure every 100 ms and 1.1 mV when using the average.

Another example of preprocessing data using the computing capacity of the slaves is the battery module temperature measurement. The BMS ASIC does not directly measure the temperature. To perform this task, the system uses a thermistor (R_{Temp}) placed inside the battery module, which is connected to a measurement circuit in the slave (Figure 5.2). Then, the BMS ASIC measures the voltage between the terminal of the thermistor to estimate its resistance using the voltage divider circuit equation. Finally, it retrieves the module's temperature with the relation between resistance and temperature of the thermistor, provided by the manufacturer. In a traditional BMS, these calculations are made by the central device for all the modules. However, in our WBMS implementation, it directly receives a temperature measurement signal.

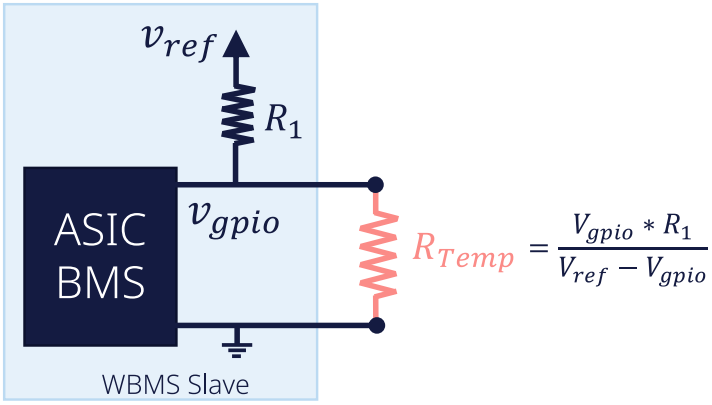


Figure 5.2 – Typical approach for battery module temperature measurement.

In the following sections of this chapter, we present and test two more complex tasks that we propose to be embedded in the WBMS slaves. The first one concerns the cell balancing voltage offset compensation which would increase the effective time of discharge. The second shows how it is possible to estimate the SOC of the battery using the slave's computing capacity.

5.1 Cell balancing compensation

When the cells of the battery pack do not have the same charge level, the BMS must discharge the battery cells with the highest level until all the cells have the same charge. Typically, the BMS ASIC are prepared for this operation, as they include some

FET that enable/disable the discharge of the cells, as presented in Section 2.1.3. The main drawback of this function is that it shares part of its circuit with the measurement function, so the cell discharge produces a perturbation in the voltage measurement. To understand the type of perturbation, Figure 5.3a shows an example of the equivalent circuit of three battery cells connected in series (V_1, V_2, and V_3) with internal resistance R_{i1}, R_{i2}, and R_{i3}. These cells are linked through some wires with resistance R_{w1}, R_{w2}, R_{w3} and R_{w4} to a BMS slave. The slave has a balancing circuit composed of some resistors R_b, capacitors C_b, and a switch (FET) for each cell. There is also the voltage measurement circuit (represented in color green) which has an RC circuit to filter some noise before the ASIC measure in the ports V_{c0}, V_{c1}, V_{c2}, and V_{c3}. Observe that if all the CB switches are open, there is no current flowing in the circuit, and the cell voltage is easy to measure; the voltage of cell 2 (V_2) would be equal to the difference between the voltages measured by the BMS slave in ports V_{c2} and V_{c1}.

On the other hand, when the BMS begins the discharging process in one cell, as in Figure 5.3a with cell 2, i_2 is not zero anymore, and we would have:

$$\begin{aligned} V_{c1} - V_{c0} &= V_1 + i_2 * R_{w2} \\ V_{c2} - V_{c1} &= V_2 - i_2 * R_{w2} - i_2 * R_{w3} - i_2 * R_{i2} \\ V_{c3} - V_{c2} &= V_3 + i_2 * R_{w3} \end{aligned} \tag{5.1}$$

Where:

$$i_2 = \frac{V_2}{R_{i2} + R_{w2} + R_{w3} + 2 * R_b} \tag{5.2}$$

As you can observe, $V_{c2} - V_{c1}$ is no longer equal to the measure of V_2, and its value decreases because it is affected by the current (i_2) interaction with the wire and the internal resistance. Also, the measure of the other two cells increases as they share a connection wire with their neighbor. Figure 5.4a shows the measured voltage value in our WBMS system with the TI BQ79616 ASIC for one minute with the balancing active in three cells alternating every 5 seconds between odds and even cells. In this case, we can evidence the voltage cell offset described by Equation 5.1 in the voltage measurements when the cells discharge is active.

In a typical application, when the BMS needs to measure the cell voltages during the cell balancing process, it pauses the discharge mechanism to avoid the perturbations presented in Figure 5.4a. In this approach, the necessary time to take a new measure

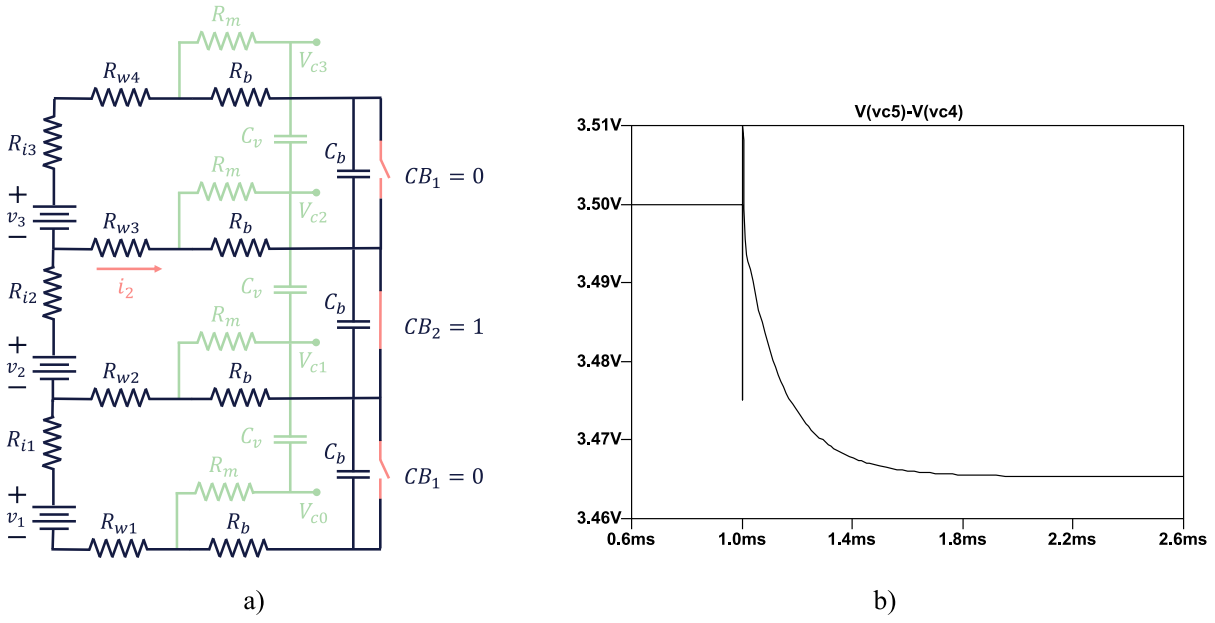


Figure 5.3 – a) Equivalent circuit of three battery cells connected to a slave, which contains a balancing and measurement circuit. b) LTspice cell 5 voltage transient response result when the balancing switch is closed at 1 ms.

depends on the RC filter used in the slave. To know how much time we need to wait before resuming the balancing, we simulated the equivalent circuit presented in Figure 5.3a in LTspice for eight cells and the component values of our WBMS slaves. Figure 5.3b presents the measure voltage plot for cell 5 when the switch balancing was enabled at 1 ms from the beginning of the simulation. The simulation indicates that the voltage signal needs almost 1 ms to reach its final value. Using this approach, in our WBMS solution, the slave must pause the cell balancing for 1.5 ms (settling time + measure time) at each time it needs a new measure (every 20 ms, as presented in the introduction of this chapter). If it does not respect this wait time, the measured voltage would not be the expected continuous signal but an input with some perturbations. To illustrate this phenomenon, Figures 5.4b and 5.4c show the voltage plot while the balancing was active in three cells for one minute with a pause of 1 ms (0.5 ms + 0.5 ms) and 1.5 ms respectively.

Pausing the cell balancing during 1.5 ms every 20 ms would lead to losing 7.5 % of effective discharge time (75 ms per second). In 8 hours, which is a typical balancing duration, it would mean 36 minutes lost. To reduce the balancing pause intervals, we propose an algorithm to compensate for the voltage offset caused by the discharge operation. A

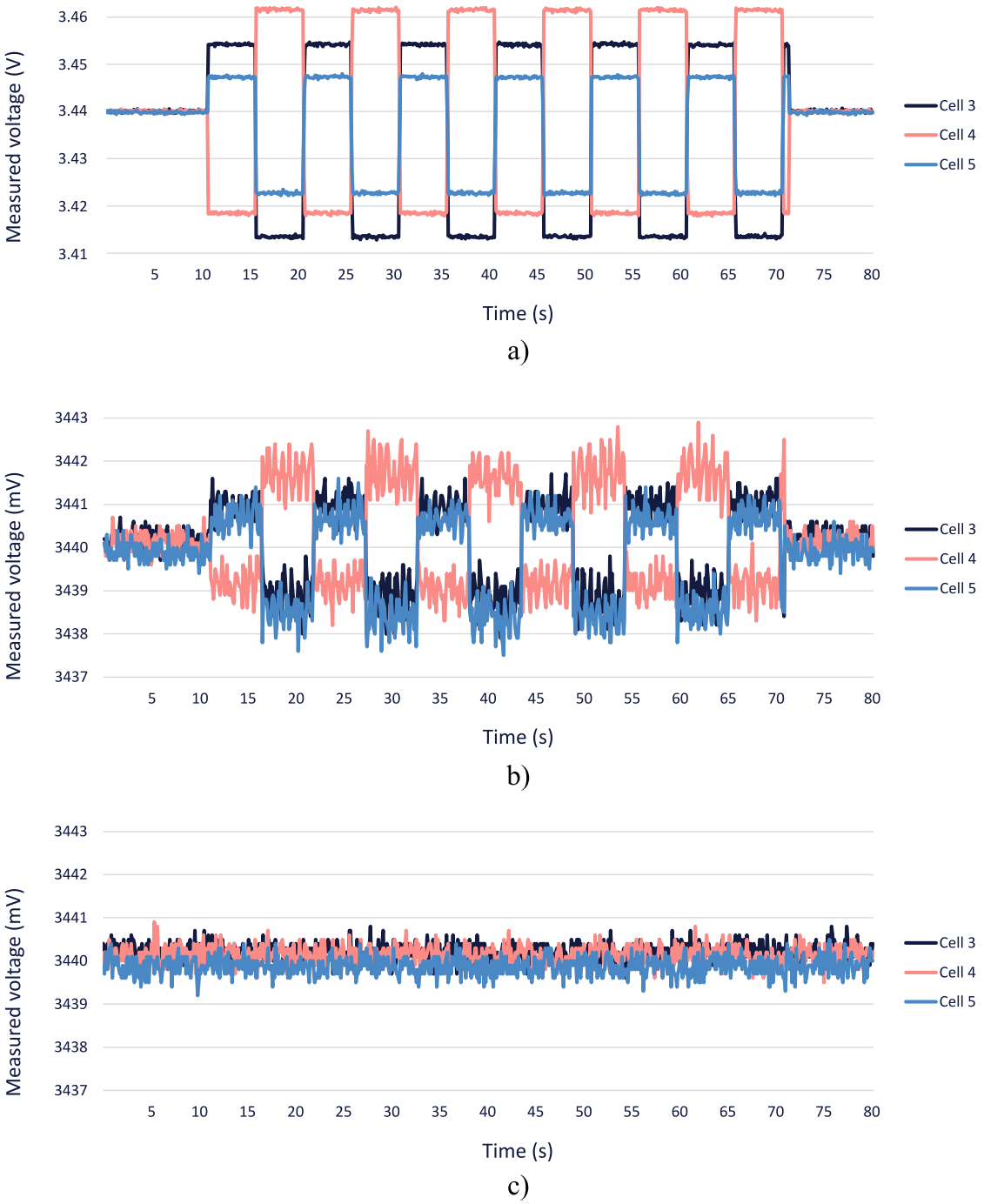


Figure 5.4 – Battery cell voltage measure for three cells when the balancing is active. For each case the data was collected during 80 seconds with 10 seconds at the beginning and at the end without balancing enabled, **note that not all the plots have the same y-axis scale**. a) Voltage data without pausing the balancing at any time. b) Voltage data with 1 *ms* of pause before executing the measure. c) Voltage data with 1.5 *ms* of pause before executing the measure.

possible alternative would be to use the equivalent circuit and the equations presented before. This method would need to know all the resistance values, which is not a problem for the R_b resistors because they are components chosen in the slave design. The internal resistance is not easy to estimate, but in a typical application, its value is minimal compared to the other resistances, so it is possible to ignore it. The wire resistance depends on several factors like the length of the wires, the fabrication and assembly process of the module, the battery temperature, or the state of the connectors. This dependence makes it difficult to establish a fixed value for all the battery cells, so it is necessary to use an algorithm that continuously estimates the equivalent resistance.

Using the measure of the battery cell voltage before enabling the discharge process and after it was enabled gives us the voltage offset produced by the balancing current, which allows us to find the resistor values with equations 5.1 and 5.2. Then, we can estimate the cell voltage values even with the balancing enabled. However, this method requires several calculations that are not necessary. Because this method strongly depends on the voltage offset measured by the ASIC, we propose only using this value to compensate the voltage measures, as described in algorithm 2. With this algorithm, we can reduce the pause intervals and avoid circuit-based calculations. Our algorithm proposes to use two tasks; the first one loops every T_{update} to pause the cell balancing, measure the cell voltage and resume the discharge operation. Then, the second task measure a new value every $T_{measure}$, computes the voltage offset when the other task indicates it, executes the voltage compensation, and, every $T_{refresInterval}$, computes the average to send a new data to the root. It is essential to ensure that T_{update} is a multiple of $T_{measure}$ so the interval between the V_{ref} measurement and the V_{offset} computation is as short as possible. In this way, the voltage changes produced by the current fluctuations would not affect the algorithm's performance.

We implemented this solution in our WBMS for the Renault Zoe battery pack to validate it. In this case, each slave updates the voltage offset every 500 ms, takes a new measure every 20 ms, and sends new data to the root every 100 ms. Figure 5.5 presents a box plot comparison between the voltage measure obtained with our proposed method, the classic approach, with a wait time of 1.5 ms (Figure 5.4c) and 1 ms (Figure 5.4b). For all the cases, we present the voltage of the same battery cell when the balancing was active for one minute in the pack (balancing alternation every 5 seconds between odds and even cells). As you can observe, our proposed method compensates for the voltage offset of approximately 20 mV observed when there is no pause (Figure 5.4a). The compensation

Algorithm 2 : Voltage measure algorithm during balancing with offset compensation

while True **do** ▷ Task 1 loops every T_{update}
  Pause balancing
  Wait settling time
  $V_{ref} \leftarrow V_{measure}$ ▷ Measure without balancing for each cell
  $newref \leftarrow 1$
  Resume balancing
end while

while True **do** ▷ Task 2 loops every $T_{measure}$
  $V_{cell} \leftarrow V_{measure}$ ▷ Measure with active balancing
  if $newref = 1$ **then**
    $V_{offset} \leftarrow V_{ref} - V_{measure}$ ▷ Voltage offset update for each cell
    $newref \leftarrow 0$
  end if
  $V_{cell} \leftarrow V_{cell} + V_{offset}$ ▷ Voltage compensation

  if send data **then** ▷ Every $T_{refreshInterval}$
    Compute average
    Send voltage measure to WBMS root
  end if
end while

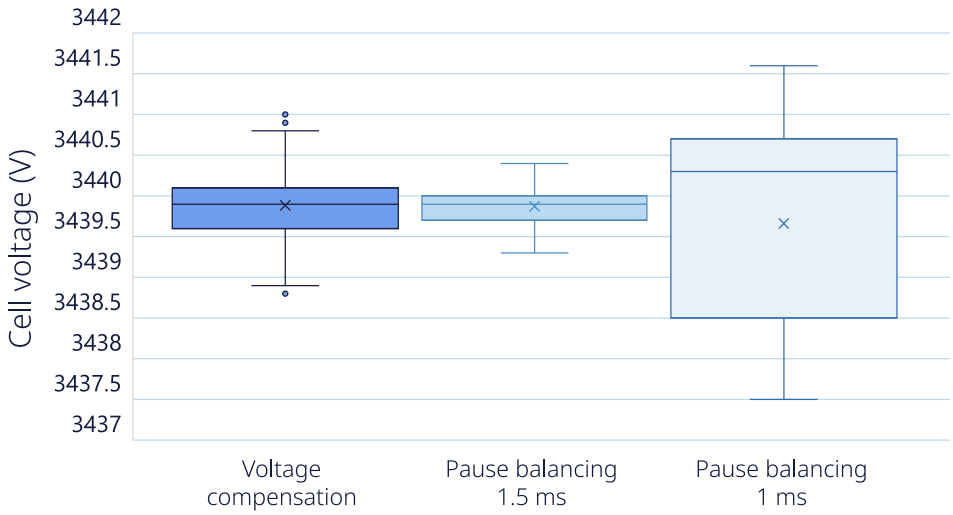


Figure 5.5 – Comparison of the voltage measure precision when the balancing is enabled between our proposed compensation algorithm and the classic approach, which requires pausing the discharge operation each time a new measure is required.

algorithm returns values within an interval of 2 mV, which is very close to the classic approach's performance (1 mV) with a wait time of 1.5 ms. The main advantage is that the system only needs to pause the balancing during 3 ms per second (0.3 %) with our proposition. Additionally, with the WBMS distributed capacity, there is no necessary intervention of the master to execute this task.

5.2 Distributed battery SOC estimation

The SOC estimation is very useful for understanding the status of the battery. Based on this value, the BMS can set a limit on the input or output power, calculate the remaining autonomy of the vehicle, or the necessary recharge time. Typically, a BMS models the battery pack as a single cell and computes only one SOC for this average cell. The Renault BMS team prefers to estimate the SOC of every cell and store it over time, as it is helpful to obtain a good estimation of the battery cells' health. With these data, the EV could identify when a cell is no longer suitable for an automotive environment. Additionally, if each module keeps this information for a second life application, it would be possible to know more precisely the properties and limits of the cells. Estimating all the battery SOC periodically (e.g., 96 cells every 100 ms) in a single central microprocessor requires considerable computing power, which would increase the costs of the BMS and its energy consumption. In this section, we show the capabilities of the low-power microcontrollers used for the WBMS communication to execute the SOC calculations.

Section 2.1.3 explains some options to compute the SOC of LIB cells. We have seen that Kalman filtering is a good option for dynamic systems as it has mechanisms to adjust the estimation deviations caused by sensors' measurement noise. This algorithm needs a model to predict the state (SOC) of the system (cell), which depends on the materials of the cell and requires a characterization process. The Renault Zoe traditional BMS uses this Kalman filtering in a centralized manner, so the Renault BMS team has already created the cell model and validated the algorithm on the road for several years. In the framework of this thesis project, we had access to their Simulink simulation file to generate a C code version for the TI CC26x2 microcontroller family using the MATLAB Simulink Coder tool. We set up a TI CC26x2 Launchpad (Figure 3.6) with the SOC estimation code and verified that the output values of the algorithm were the same as in the Simulink version, using the same entries for both platforms. Once we had proved that the algorithm was working as expected in the low-power RF microcontroller, we added it

to our WBMS implementation. The function requires three inputs: voltage, temperature, and current. The WBMS slaves already have the first two as they oversee their measure using the BMS ASIC. The current value is the same for all the battery pack cells as they are connected in series and should be delivered by the network's root.

We decided to modify our scheduling function to ensure that the computing task does not impact the wireless network's performance. Figure 5.6a presents the new version where our slotframe still has 30 timeslots, but we do not use the last two for any data exchange to give the slaves the necessary free time to compute the eight battery cells SOC. Additionally, the WBMS root must send the battery pack current value using the GACK frames of timeslots 13 and 14, the way it retrieves this value is described in the next chapter. To illustrate the time repartition of the WBMS slaves functions within a slotframe, we setup a GPIO as an output which sets a high value when the slave executes an RF data exchange, retrieves a new voltage/temperature value from the BMS ASIC, computes the voltage average and executes the SOC update. Then, we measure the GPIO behavior with an oscilloscope in nodes one and two (Figures 5.6c and 5.6b). We can observe that both nodes wake up to receive the beacon, send messages in their dedicated timeslot, and receive the GACK frames. They also measure at the same time the voltage of their battery cells in the M, AV, and SOC events of Figure 5.6. In the AV mark, the nodes compute the average of the last five measures, as described in the introduction of this chapter. Finally, in timeslot 27, the nodes have three timeslots with no activity to estimate the SOC of all their battery cells using the average computed in the AV event and the current value received in the GACK frame from the root. The node can complete this task in 6 ms, which makes it possible to obtain a SOC update of all the 96 cells every 100 ms using the distributed capacity of the WBMS and keep the robustness of the wireless network.

5.3 Conclusion

Throughout this document, we have discussed the multiple advantages of introducing a wireless link between the master and the slaves in a BMS, such as the reduction of the design complexity of the battery pack, the space gained to enhance the battery capacity or the reduction of connector failure risks. Managing a wireless network communication requires the inclusion of a low-power microcontroller in each node of the network because it needs to keep synchronization and process the data input/output. The presence of this

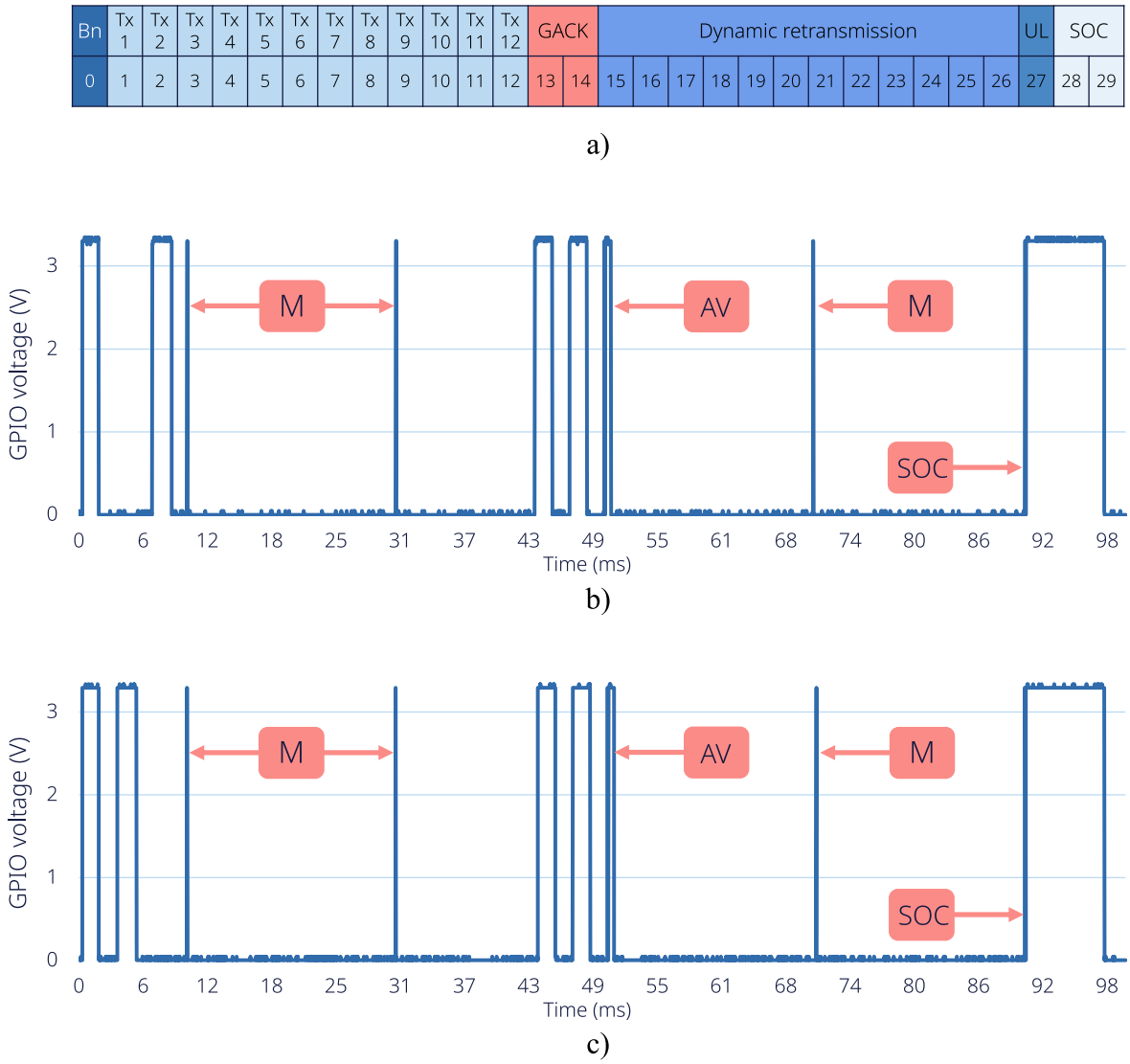


Figure 5.6 – a) New TSCH schedule for our Renault Zoe WBMS network implementation where the last two timeslots are dedicated to the SOC estimation. b) GPIO measure for the node 2. c) GPIO measure for node 1. For b) and c) the GPIO is high when the node executes an RF data exchange, retrieve a new voltage measure, compute the voltage average and update the SOC estimation. Both nodes executes five voltage measures during one slotframe in events M (M = measure), at the beginning of event AV (AV = average) and at the beginning of event SOC. In the AV event the node also computes the average of the last five measures. In ther rest of the SOC event, the node computes the state of charge of the eight battery cells, it takes around 6 ms.

additional computing power in each device of the WBMS opens the door to a new chapter in the BMS software architecture to enhance the battery management algorithms, which would allow better use of the battery capacity. In this chapter, we first presented simple operations that the slaves can execute instead of the master, such as the synchronized voltage measure every 20 ms, the temperature estimation, or the cell balancing management. In addition, to reduce the workload of the master, it is possible to embed the BMS ASIC driver in the slaves and make it transparent for the BMS master. This would allow the system designers to change the BMS ASIC without having to modify the software of the BMS master. We also showed the possibility of improving the efficiency of the cell balancing mechanism through our offset compensation proposal, which increases the battery cell discharging time. Finally, we presented the process of including a Kalman filtering algorithm in the low-power microcontrollers to allow the system to update every 100 ms the SOC estimation of all the battery cells. This can be seen as the first step of a "smart module" approach, where a battery module will include a WBMS slave during its lifetime, which can store the cells usage history and estimate the SOC or SOH without the intervention of a central unit.

In this chapter, we have complemented the WBMS network design with some additional features for the WBMS application layer, which take advantage of the additional computing capacity. We can now test our system in the Renault Zoe and execute some tests outside the laboratory to verify the performance of our proposal in a real environment.

Chapter 6

WBMS VEHICLE TESTS

In the previous chapters, we discussed the advantages and challenges of introducing a wireless network in the BMS, a critical system. From the theoretical discussion to the implementation in real hardware, we have shown the development of our proposed solution to fulfill all the requirements of the WBMS application. The proposed mechanisms have been tested in the laboratory using a Renault Zoe battery pack to assess its performance. In this chapter, we will take our implementation to the next level and put our system inside an EV to validate that the proposal presented in this thesis works properly in different real environments.

Before mounting the battery pack in the vehicle, some adaptations must be made to add a Controller Area Network (CAN) interface to the WBMS root. In this way, we will be able to retrieve the network data in the cabin once the network is sealed under the vehicle. The adaptation process is described in Section 6.1. Then we study the possibility of using a WBMS master placed outside the battery pack in Section 6.2. Next, we present some tests done to verify how exposed the network is to external interference in Section 6.3. Finally, in Section 6.4, we summarize the results obtained in the chapter and present our conclusions.

6.1 WBMS setup for EV tests

Before mounting the battery pack in the vehicle, it is necessary to create an adequate setup for the tests as we would not have physical access to the WBMS devices. It must be possible to retrieve the battery status, the network statistics, the current measure, send commands to our wireless system, and update the firmware of the devices. We added a CAN interface to the WBMS root to achieve all these tasks, a very common wired communication protocol for automotive applications. The main reason to do this is that the current battery pack sensor communicates with the BMS over a CAN bus where we can plug the WBMS root to receive the current measure and send it to the network in

the GACK frames. Also, this CAN bus comes out from the battery pack through a data port to which we can connect an extension cable and route it to the vehicle cabin. Once in the cabin, we can use a USB to CAN adapter to interact with the WBMS master using a laptop or a Raspberry Pi.

Texas Instrument offers a CAN plug-in module[1] which is compatible with the TI cc26x2 Launchpad and communicates with the wireless microcontroller over Serial Peripheral Interface (SPI). We set up the WBMS root to send every 100 ms several 8-byte CAN messages containing:

— 96 cell voltages measures and 12 module temperature measures, each one coded in 16 bits.
— 96 16-bits SOC estimation values.
— Cell balancing status of all the battery cells.
— The number of messages lost before and after retransmission.
— The radio channel used for each reception and the RSSI value.

We also added the possibility to update the firmware of the network using the CAN interface (Firmware Over the Air – FOTA). This way, we can improve our prototype continuously during the test phase without needing to extract the battery pack from the vehicle.

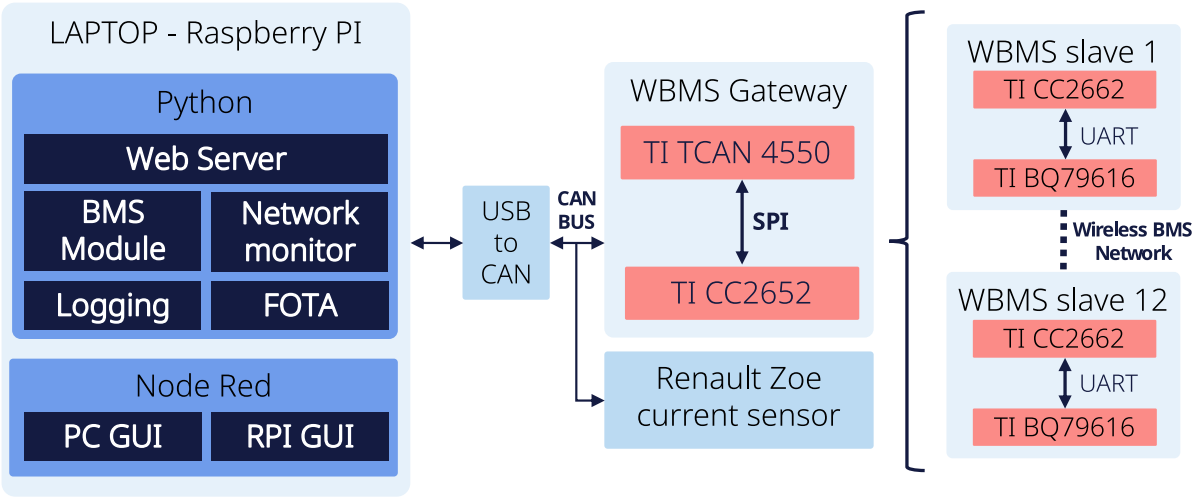


Figure 6.1 – Proposed WBMS setup test with CAN interface. The WBMS root uses the CAN interface to receive commands and the current measures. It returns the data received from the nodes and the statistics of the network to analyze its performance during vehicle tests.

1. https://www.ti.com/tool/boostxl-canfd-lin

To interact with our WBMS over CAN, we developed a python script that receives, processes, and logs the battery cell measures and network statistics. We included the option to send the necessary commands to switch between the active and sleep mode of the BMS. The script also manages the FOTA process and shows in real-time its progress. Finally, we added a GUI interface using a python web server and Node-RED for demo purposes. Figure 6.1 shows the general architecture of our complete WBMS with CAN interface and details the communication protocol used between the different hardware elements. Once the system was ready and tested in the laboratory, we mounted the battery pack in the vehicle. Unfortunately, we had some problems, and we lost node eight during the first tests, which was impossible to recover unless we unmounted the battery pack. Extracting the battery pack from the EV is not easy, so we preferred to test the system without this node as the rest of the network was working correctly.

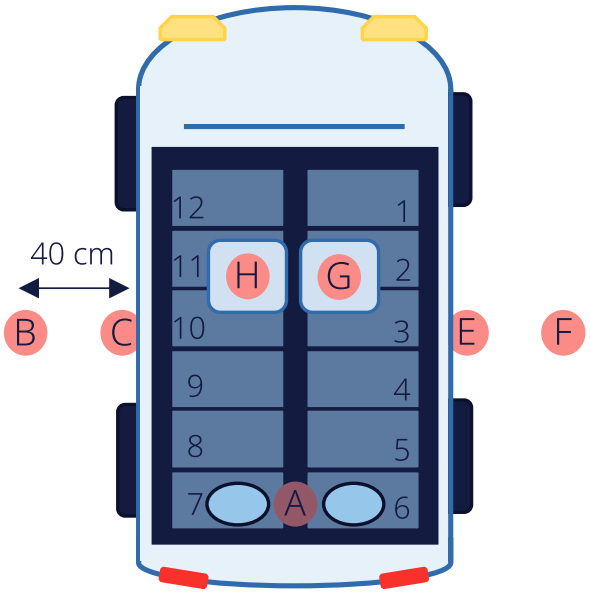


Figure 6.2 – WBMS master positions for signal strength experiment with the 12 WBMS slaves inside the battery pack connected to each module. A) Default position, which is the only one where the master is inside the battery pack. B) 40 cm to the left of the vehicle. C) Next to the vehicle between the left wheels. D) Under the vehicle in the center (not shown). E) Next to the vehicle between the right wheels. F) 40 cm to the right of the vehicle. G) Passenger's seat. H) Driver's seat.

6.2 External root signal strength

We are interested in measuring how exposed are the electromagnetic signals of the WBMS to the outside world for two reasons. The first one is that a possible advantage of the WBMS is that we would be able to diagnose, update and repair the system without needing to unmount the battery pack from the vehicle. Second, these results would help estimating the security risks that the network may face. In this test, we measured the RSSI of the messages received by the master from each slave for different positions of the master outside the battery pack. We then compare the results to the default case when the master is inside the battery.

Figure 6.2 presents the positions where we have placed the master of the network during the tests. We placed the root on the floor $40cm$ away from the vehicle on both sides ($B - F$), under the vehicle between the wheels, and in the center ($C - E - D$). We also test it by placing the master in the driver and passenger seats ($G - H$). For each position, the master collected data for one minute, and then we computed the average RSSI value for each node. The results are plotted in Figure 6.3 and compared against the default case where the master is inside the battery pack (A). The RSSI values of -90 dBm indicate that the master did not manage to connect with these nodes during the test.

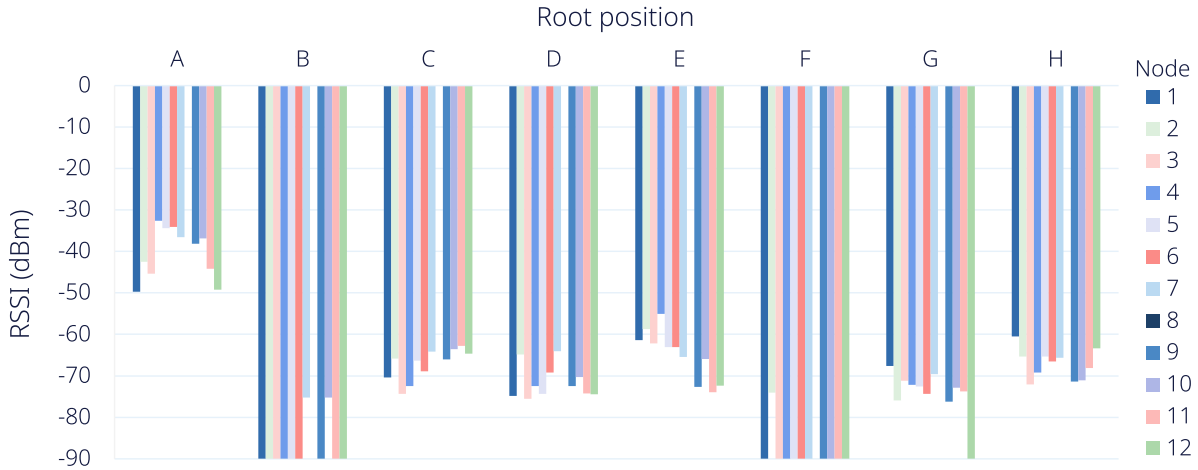


Figure 6.3 – Average RSSI values of received messages from each node when the master is placed in the positions described in Figure 6.2. Position A is the reference case where the master is placed inside the battery pack.

As you can observe, accessing the network when the master is outside the battery pack is possible. The signal strength is much weaker than the default case, which is expected

as we have some metallic obstacles in the middle. When the master was 40 cm away from the vehicle, it was not possible to reach most of the nodes, and the one that it managed to find had a very poor signal strength. For positions C and E, we observe that the nodes placed on the corresponding side of the battery pack present a stronger link than the others. During the test, we were forced to update the nodes' firmware with the master outside the vehicle. We found that the best way was to update the nodes of one side using position C and then the others by putting the master in position E. Inside the cabin, it was also possible to reach the nodes but with slightly poorer performance. Now that we measured the ability of the WBMS signals to pass through the metallic barriers of an EV, it is time to study how external interference affects our wireless solution for the BMS.

6.3 External interference evaluation

Even though in a WBMS application, the network is enclosed by a metallic structure that protects against external interference, we want to evaluate the system's robustness when the battery pack is mounted in the vehicle. In this section, we present the results of several tests where we measured the channel stability of the network in the presence of external interference to estimate the vulnerability of our proposed WBMS.

For the first tests, we generated the interference using two Wi-Fi routers and four computers which exchange data using the SCP network protocol, in the same way we did for the channel blacklisting mechanism (Section 3.3.3). For the test presented in this section, we disabled the channel blocking because we do not want the network to stop using the channels with poor performance, as it would significantly impact the results. To understand the effect of the metallic enclosure and the position of the Wi-Fi transceivers, we compare the following setups:

1. The battery pack is placed in the laboratory without the cover, and the Wi-Fi transceivers are located 1 meter away from the system.
2. The same setup as in the previous test, but the battery pack has its cover in place in this case.
3. The battery pack is inside the vehicle, and the Wi-Fi transceivers are inside the cabin. Two computers and one router are in the driver and passenger seats. The other elements are placed in the back seat of the vehicle.
4. Battery pack inside the vehicle and Wi-Fi transceivers outside on the floor in positions C and E (Figure 6.2) with the doors closed.

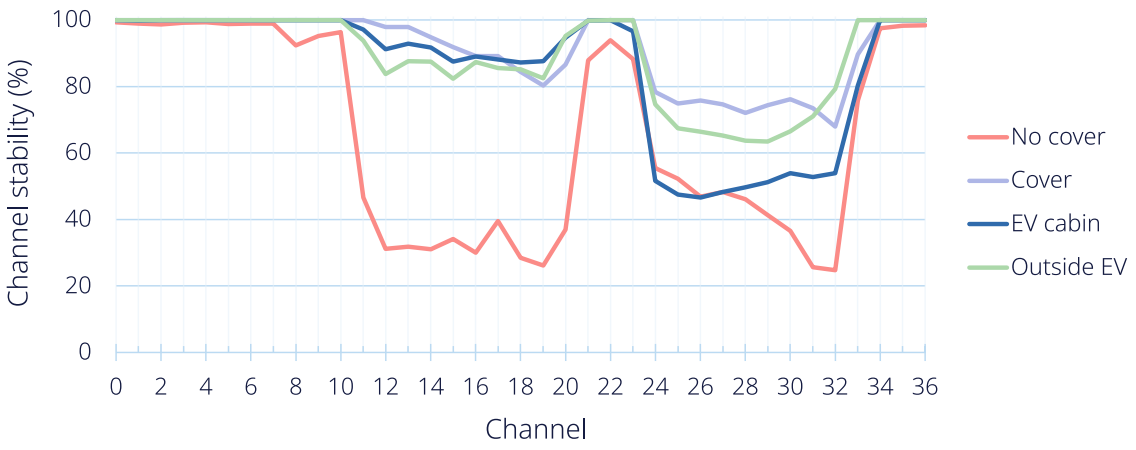


Figure 6.4 – WBMS network channel stability comparison for the four setups described before with external interference generated using two Wi-Fi networks and constant traffic.

Figure 6.4 shows the channel stability results of the tests. For each test, the master received data for 45 minutes. As you can observe, the worst case is when the battery pack does not have its cover in place as there are some channels where the stability went below 30 %. Then, we noticed that the test in the laboratory with the cover in place and the test with the Wi-Fi transceivers on the floor outside the vehicle showed similar results. The EV panels provide extra protection against external interference because we must put the transceivers just next to the vehicle rather than 1 meter away to produce a similar effect when the pack is in the vehicle. For both cases, the network lost 25% fewer messages before retransmission than in the first test case. Inside the cabin, we obtained some very interesting results, the Wi-Fi network placed in the front seats (channel 6) did not produce a significant impact compared to the previous cases. However, the network in the back seat (channel 11) makes the stability of the corresponding BLE channels go down to 50%. This is because the Renault Zoe battery pack is air-cooled, and the air entrances are two holes in the top of the battery pack cover just above the WBMS master (Figure 6.2 next to position A). When the battery pack is inside the vehicle, the holes are placed under the back seat, which is an easy entrance for external interference. Table 6.1 condensate the test results, which prove that even in this application where the network has metallic protection, it is still exposed to external perturbations. So, a channel blacklisting mechanism must be used to keep highly reliable communication in every situation.

We also wanted to confirm that the EV's electromagnetic interference generated by the power train does not interfere with the WBMS network. In Section 2.1.1, we reviewed

	Case 1	Case 2	Case 3	Case 4
Total messages	324 000	324 000	297 000	297 000
Lost before retransmission	111 282	28 987	45 045	33 071
Lost after retransmission	24 124	8	6	6
Reliability before retransmission	65.6534 %	91.0532 %	84.8333 %	88.8648 %
Reliability after retransmission	92.5543 %	99.9974 %	99.9776 %	99.9780 %

Table 6.1 – Comparison results between the four setups created to test the WBMS network in an environment with high external interference.

some works that affirmed that this interference was more important in a frequency band much lower than the 2.4 GHz of the BLE physical layer used in our case. We executed a 40-minute driving test to confirm this and measure the network's performance. The results show no additional perturbation caused by the power train. In fact, from 318 802 messages transmitted, 123 were lost before retransmission. This results in a reliability of 99.9614 % before retransmission, which is very similar to the results presented in Section 3.3.2 where we tested the battery pack in the laboratory with its cover in place. We also analyzed the RSSI for each node over time, and we did not notice any change when the current of the battery pack was different from zero. During this test, the final reliability after the dynamic retransmission was 100 %, the same performance obtained in the laboratory tests.

6.4 Conclusion

The main objective of this chapter was to test our proposed WBMS solution outside the laboratory in a real environment. The first step was to adapt the WBMS root with a CAN interface to interact with it and receive all the data collected from the network. We also added a FOTA capability to our system, allowing us to update the nodes' firmware without needing to unmount the battery pack from the vehicle. Once the system was ready, we installed the battery pack in a Renault Zoe. In the first test campaign, we put the master of the network in different positions outside the battery pack to see if it was achievable to communicate with the nodes. The results show that it is possible depending on where we placed the root. In general, when the master is more than $40cm$ away from the vehicle, the root did not manage to reach the network. As expected, the signal strength was much lower compared to the case where the manager was placed inside the battery

pack. These results show that it would be possible to connect an external agent to the network outside the vehicle to obtain the network status, update the firmware, or even diagnose the system in case of failure. Note that in our tests, we use a master based on a TI cc26x2 Launchpad board with a printed antenna. The quality of the connection can be improved if we use hardware with a different antenna configuration with better sensitivity. However, these results indicate that even if the network is enclosed inside a metallic structure, from a security perspective, it should be designed in the same way as if the barrier did not exist.

We then tested the network stability in the presence of external interference generated by some Wi-Fi routers and four computers using the SCP protocol to generate constant traffic. For these experiments, we compared different configurations to understand the impact of the interference transceiver position and the metallic enclosure. We have constated that placing the battery cover and putting the pack inside the vehicle reduces the impact of the external interference by 25%. We also saw that if the battery pack cover has cooling holes and we place the transceivers inside the cabin close to these openings, the channel stability of the network is highly deteriorated. This depends on the battery pack design and must be considered if using a WBMS. New vehicles have multiple info entertainment systems, which may produce heavy electromagnetic interference in the 2.4 GHz band. These results have shown that the metallic enclosure does not protect the network entirely from external interference, so using a channel blacklisting method is necessary to keep the high reliability of the system.

Chapter 7

Conclusion and perspectives

7.1 Conclusion

The automotive industry is in the middle of a transition process. For over 100 years, they were dedicated to producing Internal Combustion Engine (ICE) based vehicles. Today, from an environmental and legal point of view, ICE vehicle production must stop, and EVs are seen as the best replacement. Even though EVs appeared before ICE vehicles, only until the beginning of the XXI century their development and production experienced a massive acceleration. Nowadays, EVs are an active subject of research as manufacturers try to democratize electric mobility by optimizing and reducing costs everywhere. The most critical element of an EV is the battery. An EV battery pack is composed of cells grouped into modules; for example, the Renault Zoe battery pack has 96 cells and 12 modules. At the moment of writing this document, most of the EVs on the road used Lithium (LI) based battery cells. This kind of receptacle has many advantages, and without any doubt, they are responsible for making electric mobility a real option to replace ICE mobility. However, they are required to operate in the optimum temperature and voltage conditions; otherwise, the vehicle's safety is not guaranteed. For this reason, every vehicle uses a Battery Management System (BMS), a set of sensors and actuators in charge of monitoring and ensuring the correct operation of the battery pack. In a classic approach, the BMS uses a master-slave architecture with a wired daisy chain link between them. This dissertation aims to propose a solution to replace the wired BMS interface with a wireless link. It can bring several advantages like an easier battery pack design, saving space, reducing connector failure risks, or enabling full automated battery pack assembly.

In Chapter 2, we presented the necessary background and state of the art to understand the challenges of this Wireless BMS (WBMS) project. We explained the typical EV architecture, the basics of Li-based batteries, and the primary function of a BMS. From this review, we conclude that it was necessary to implement a network with high

reliability, bounded latency, and low energy consumption to achieve a similar level of robustness to traditional wired BMS. Then, we presented the state of the art of low-power wireless networks to study which options we have to implement our WBMS proposal. We reviewed the IEEE Std. 802.15.4, the standard for Low-Rate Wireless Personal Area Network (LP-WPAN), especially their LLDN and TSCH MAC modes, the Bluetooth Low Energy (BLE) networks, and the Wireless IO-Link protocol stack. We observed that even if these networks have several advantages, none of them was specially made for an application such as the WBMS. However, we saw that mechanisms like time division, frequency diversity, GACK retransmission, and channel blacklisting had shown excellent results in improving the performance of low-power wireless networks. In the last part of Chapter 2, we reviewed the two principal commercial options for the WBMS provided by TI and ADI that promise very good performance but none of them detailed how they achieve these levels of robustness.

In the rest of the document, we presented and tested our WBMS proposal. In Chapter 3, we detailed the active mode of our system where the master must receive a new measure from all the slaves periodically (every 100 ms for the Renault Zoe). We proposed a framework to create a Wireless Sensor Network (WSN) for critical applications with high reliability, bounded latency, and low-energy consumption. We designed a TSCH-based network running over the BLE physical layer. On top of that, we developed a scheduling function that uses a similar approach to the GACK method of the LLDN MAC mode to manage the retransmissions dynamically. Additionally, we created a channel blacklisting algorithm adapted to our scheduling function to keep the network's robustness even in highly perturbed environments. We implement our WBMS in a TI material specially made for the Renault Zoe battery pack to validate our proposal. After multiple tests, we probed that our method outperforms a traditional retransmission approach; it can achieve 100 % of reliability even when the network is under high external interference while keeping the latency within the required limit and with an average current consumption of 400 μA per slave. With these results, we demonstrate that it is possible to enable robust wireless communication on critical applications such as the WBMS using our TSCH-based approach.

Once the active mode of our proposal was defined and tested, in Chapter 4, we presented the sleep mode of our WBMS. When the EV is parked, a traditional BMS can turn off all the slaves because it can send an interruption over the wired link to wake them up. In a traditional low-power wireless network, there is no way to send an interruption,

making it impossible to turn off the slaves. The master must maintain the network synchronization to keep the energy consumption low. Because the master is powered by the small 12 V battery of the vehicle, this behavior may produce its total discharge. We proposed to modify the network architecture during sleep mode, where the nodes exchange beacons between them and keep the synchronization without needing to have the WBMS root awake. Our proposal allows the system to spread all the battery cell voltage measures over the network, which can be used to keep the battery pack balanced without the necessity of the central BMS. In our tests, the network stays synchronized with an average current consumption of 40 μA per slave, which is negligible compared to the capacity of an EV battery module. If necessary, the power consumption can be reduced but it would require to extend the maximum wake-up time which in our case was set to 1 second.

In Chapter 5, we discussed the new opportunities and changes for the BMS software that the introduction of a wireless network brings. In a WBMS, the slaves require a low-power microcontroller to manage the network synchronization and interact with the BMS ASIC. In other words, with a WBMS we obtain a new distributed computing capacity that we can use to execute some battery management functions. We showed that this microprocessor could execute simple tasks autonomously, like the voltage and temperature measurements. However, we also demonstrated that executing more advanced functions like our proposed voltage offset compensation for the cell balancing or even the computing of the battery cells SOC is possible. We added a Kalman filtering SOC estimation algorithm to our TI CC26x2-based WBMS implementation and showed that it only takes 6 ms for these microcontrollers to update all the SOC estimations. These results proved that the new distributed computing capacity of the WBMS can reduce the central BMS's workload and significantly influence the possibilities of second-life applications for the battery modules.

Finally, with our WBMS completed and tested, we mounted a battery pack in a Renault Zoe equipped with our prototype. To do this, we had to add a CAN interface to the WBMS root to interact with the network from the cabin using a python script. We described the process in Chapter 6 and presented the results of several tests conducted in the vehicle. The tests showed that the laboratory experiments were also valid in a real environment, as the power train interference did not influence the system. The results show that our proposed TSCH-based network with dynamic GACK retransmission management running over the BLE physical layer provides the required level of reliability with bounded latency and low-power consumption inside an EV. The tests showed that the system is

still vulnerable to external interference even if it is protected by several metallic layers of the vehicle, which makes it mandatory to include a channel blacklisting algorithm in the network. We also showed that a node outside the vehicle could reach the network, which opens the possibility to multiple applications but also has several security risks. During our test campaign, we did not observe any alteration of the other RF systems of the vehicle, such as the keyless access or the info entertainment system.

7.2 Perspectives and future work

This thesis proposes a WBMS TSCH-based network running over the BLE physical layer with dynamic retransmission management and channel blacklisting. The designed system has met the requirements of such a critical application and was tested in multiple scenarios, showing the viability of the WBMS. The adoption of this wireless application is a reality as many EV manufacturers have decided to go with it for their upcoming releases. However, some open topics are worth mentioning to use the potential of this promising new architecture for the BMS.

7.2.1 Outside world access to the WBMS

In Chapter 6, we observed a possibility to access the network from outside the battery pack depending on where we place the external node. These results show that introducing the wireless network inside the battery pack could lead to a BMS system diagnosis and repair without unmounting the battery from the EV. In our case, when we put the pack inside the vehicle, something went wrong in our system, and the nodes switched to an undesirable state. The only way to recover the system was to use a particular version of the WBMS root to update the nodes' firmware. Because the root update was impossible, we updated the nodes using an external root placed next to the vehicle. This possibility can save a lot of time and effort in the after-sale life of an EV equipped with a WBMS.

However, it also opens the door for potential security risks to the system because the signals of the WBMS are exposed to the outside world. We believe that the wireless network of such a critical system must guarantee the authenticity of all the messages as well as its confidentiality. In this way, the nodes can be sure that the messages received are from an actual system member, and no external device would be able to read or replicate their messages. The IEEE Std. 802.15.4 describes very clear procedures to implement the

necessary security mechanisms in low-power networks as it does not recommend design a system without any level of security. This topic leaves an open question of which type of methods the WBMS must use to share the necessary encryption keys between the nodes. Depending on the assembly procedures of the vehicle, the manufacturers may prefer to use pre-shared keys downloaded to the nodes just before being mounted in the battery pack, or it may be preferable to use a public-private key method instead. This topic was not aborded in this document as it was out of the scope of our work; nevertheless, we believe that a study to find the most optimal method for the WBMS series production can be helpful before commercial use.

7.2.2 Inside EV access to the WBMS

The Renault Zoe battery pack has an air-based cooling system that requires two holes in the battery cover. We had seen that placing a Wi-Fi router over these openings in the back seat created a comparable amount of interference as when the pack was in the laboratory without the cover. New electric vehicles have migrated to a liquid-based cooling system as it is more performant and allows a more compact battery pack design. They do not require holes in the metallic cover and prevent the direct exposal of the network to the interference. The results of these experiments open the possibility of using the properties of electromagnetic waves to integrate the WBMS master with the central EV microcontroller, which is generally placed outside the battery pack. Our experience in the Renault Zoe indicates that we can reach the network from outside the pack by generating the appropriate paths for the electromagnetic signals. Adding the central BMS functions to another vehicle microcontroller would reduce the amount of hardware required in the EV. However, this feature would require an additional study of the vehicle's electromagnetic signal dispersion, which can be costly and make the solution potentially unviable.

7.2.3 Second life management

The battery cells of an EV lose their performance over the years; when the SOH goes below a defined threshold, the pack must be replaced. However, the degraded battery modules can be adapted to a new application less demanding than automotive propulsion. One of the most promising second-life opportunities for batteries is large-scale stationary energy storage, where we use the receptacles to compensate for the intermittence of renewable energy. This application does not require fast charging or high output power

as in an EV, making it perfect Li-based cells with degraded performance. WBMS slave microcontrollers can trace the historical conditions of battery modules and monitor the cells throughout their lifetime. Ideally, in second-life applications, we should be able to use the same hardware and data to optimize the use of the battery cells. The possibility to change the nodes' firmware over the air would add extra flexibility to use the slaves in countless applications.

Nonetheless, to make the most of these advantages of the WBMS, it is necessary to have a certain level of standardization that does not exist now. A basic level of uniformity would be to have a common way to store the battery cell history in memory or in a cloud server where the new second-life application could find the data. A higher level would be that all the WBMS devices must be able to use a common physical layer (i.e., BLE PHY, IEEE Std. 802.15.4 PHY), so we may use modules from different manufacturers in the same storage system. There is also the possibility that vehicle manufacturers want to reserve to themselves the right to recycle the batteries and use their own second-life systems. Standardization would not be necessary for this scenario, but it would reduce the possible new applications for degraded batteries.

7.2.4 A word on WBMS safety

Safety is a central concern in the automotive world. The Renault Zoe wired BMS implements the industry's highest safety level (ASIL D). Implementing an ASIL D-capable WBMS is possible, but it is not an easy task. Texas Instrument uses the black channel concept, using the RF microcontroller and the network as a transmission medium that does not modify the data exchange between the WBMS master and the BMS ASICs. This approach is valid but makes it impossible to use the extra computing capacity of the node to execute some BMS algorithms, as we proposed in this dissertation. Another method to achieve an ASIL D system and execute calculations in the slaves would be to add a second microprocessor for each node that monitors the behavior of the main RF microcontroller. This new microprocessor will check if the results of the algorithms have sense with the measures retrieved with the BMS ASIC or if the diagnostic procedures have been executed as expected. This solution would add an extra cost that could make the introduction of a WBMS solution economically unviable. Future work on WBMS may be concentrated on studying different hardware and software architectures to achieve the maximum safety level of the automotive industry while keeping WBMS attractive from an economic point of view.

7.2.5 New regulations for sleep mode

In a traditional BMS, during sleep mode, the central unit wakes up periodically every couple of hours to monitor the charge balance of the battery pack. Up to the moment, no regulation demands extra monitoring of the battery in this mode. The proposed sleep state for the WBMS was designed under this premise as the central WBMS can wake up periodically, resynchronize to the network and check the balance status. However, if there is a regulation change or the manufacturers want to monitor the status of the cells continuously, it would be impossible to turn off the WBMS master in sleep mode completely. In this case, the system can use a simple low-power mode where the root keeps the network synchronization using a longer period and with some uplink slots where the nodes can send a default status. Another option would be to use our proposed sleep mode but adding to the master a way to be woken up by the nodes. In the state of the art, Wake-up Radio (WuR) have been developed using multiple architectures [113] [114]. The basic idea is to have an ultra-low power circuit that senses the radio medium waiting for a predefined wake-up sequence. When the sequence is detected, the WuR sends an interruption to the main microcontroller. In this way, if a slave detects an anomaly from the BMS ASIC, it sends the wake-up sequence to put the central WBMS in active mode and manage the emergency. This solution requires additional circuit elements and, therefore, an extra cost that is not always well received in the automotive industry.

7.2.6 EV and WBMS future

We are living in a revolutionary time for the vehicle industry as the EV will replace ICE-based vehicles in the upcoming years. Even though today, electric batteries allow the creation of EVs with acceptable ranges and charging times, they will continue to evolve to get better performances. With this evolution, the WBMS must change and adapt to new configurations. Introducing new battery chemistry with fewer risks of catastrophic consequences may lead to a reduction in the safety requirements of the WBMS, which would allow the creation of cheaper systems with distributed capacity. In this context, batteries with solid electrolytes (solid state batteries) would reduce the risk of thermal runaway of the current batteries; however, they are still under development. New chemistry means new BMS algorithms, and these new estimation methods would need a different BMS software architecture. For example, it may be necessary to use cloud computing to update the battery cell models continuously; in this scenario, it would be helpful to make

the WBMS packets IP compatible to communicate the slaves directly with the server.

Another revolution for the EV would be the replacement of battery modules each time they are discharged, allowing 100 % of charge within minutes. With such a mechanism, the WBMS network should be capable of accepting new nodes and adapting its scheduling to keep the required performance with multiple configurations. A modern vehicle includes several wired networks between microcontrollers, ASICs, sensors, and actuators. The WBMS is the first WSN implementation inside the vehicle power train, demonstrating that it is possible to create robust automotive systems using wireless networks. This system would open the door to future work on creating more extensive wireless networks, including additional sensors and actuators from multiple vehicle systems.

Appendix A

Resume Long

A.1 Introduction

Les véhicules électriques (VE) ont été un sujet actif de recherche et de développement au fil des ans, car ils sont considérés comme le présent et l'avenir des transports. On peut penser que l'histoire du transport électrique est relativement courte. Cependant, le premier moteur électrique à courant continu (DC) date de 1832 et les chariots électriques ont été testés pour la première fois à Saint-Pétersbourg, en Russie, en 1880. Le premier tramway régulier a été installé dans une banlieue de Berlin en 1881, la même année que l'ingénieur français Gustave Trouvé a construit le premier VE [1][2]. Au début du 20e siècle, les véhicules électriques étaient considérés comme un concurrent sérieux pour le transport routier. En 1900, 38 % du marché automobile américain était capté par les véhicules électriques [3]. Par rapport aux véhicules à moteur à combustion interne (ICE), les véhicules électriques étaient plus fiables, moins polluants et démarraient instantanément sans avoir besoin d'une force externe. Malheureusement, il n'existait aucun moyen abordable de stocker l'énergie nécessaire pour parcourir de longues distances. Lorsque les carburants pour véhicules thermiques sont devenus accessibles à tous, les véhicules électriques sont passés au second plan.

Au milieu de ce siècle, les trains électriques ont été développés dans le monde entier car l'ensemble du système était remarquablement fiable, et ils n'avaient pas les problèmes de stockage des VE car ils utilisaient des lignes aériennes pour alimenter leurs moteurs. D'autre part, les véhicules électriques ont joué un rôle secondaire et n'ont été utilisés que dans de petites applications où la pollution et le bruit seraient inacceptables, comme les entrepôts ou les terrains de golf [1]. En décembre 2015, l'accord de Paris a été adopté par la conférence de Paris sur le climat et ratifié par l'Union européenne en octobre 2016. L'accord est un cadre mondial pour limiter le réchauffement climatique en dessous de 1,5 1.5 $^{\circ}C$ et éviter un changement climatique dangereux [5]. L'un des objectifs de cet accord est d'atteindre le zéro net à effet de serre d'ici 2050, ce qui signifie réduire les émissions de

gaz à effet de serre au plus près de zéro, de sorte qu'il devrait être possible à l'atmosphère d'absorber le reste. Dans ce scénario, la mobilité électrique devrait être essentielle pour atteindre les objectifs de l'accord.

Tout au long de l'histoire des véhicules électriques, le principal aspect où ils ont perdu le combat avec les véhicules thermiques était l'autonomie. Sans aucun doute, le développement de batteries électriques capables de stocker l'énergie nécessaire pour permettre aux VE de parcourir une distance acceptable a été le déclencheur pour les placer au centre de la révolution des transports dans les années à venir. Aujourd'hui, presque tous les véhicules électriques sur la route utilisent des batteries au lithium. Ce type de réceptacle présente de nombreux avantages, comme sa densité de puissance. Cependant, ils ont également des comportements dangereux si leurs conditions de fonctionnement (tension et température) sortent des limites autorisées. Pour cette raison, chaque véhicule électrique nécessite un système de gestion de batterie (BMS). Une bonne mise en œuvre de ce système peut éviter des accidents catastrophiques et prolonger la durée de vie de la batterie. Aujourd'hui, le BMS utilise une topologie distribuée où l'unité de traitement et les capteurs sont séparés dans une architecture maître-esclave. Les esclaves contiennent un ASIC BMS pour mesurer la tension et la température des cellules, tandis que le maître a le CPU, qui collecte les données et effectue les calculs. Le système utilise une interface robuste câblée en série pour relier le maître et les esclaves afin d'assurer une bonne performance dans un système aussi critique.

Dernièrement, l'industrie s'est intéressée à la suppression du bus filaire et à la mise en place d'une liaison sans fil entre le maître BMS et les esclaves. Même s'il n'est pas habituel de s'appuyer sur la communication sans fil dans les applications critiques, la version filaire présente des inconvénients importants. Par exemple, le faisceau de câblage en cuivre est lourd et occupe de l'espace à l'intérieur d'un bloc-batterie qu'il serait préférable de remplir avec des cellules de batterie [9]. Cela réduit également la flexibilité de conception de la batterie et augmente sa complexité, ce qui entraîne un temps (argent) supplémentaire consacré à la phase de développement. Nous devons également considérer que même si le protocole filaire est robuste, le système doit utiliser des connecteurs entre les appareils qui peuvent souffrir de défaillance mécanique [9]. Un autre inconvénient serait que la complexité du harnais rend très difficile l'automatisation complète du processus d'assemblage du pack batterie [9]. Toutes les raisons précédentes ont ouvert la discussion sur le remplacement de la liaison filaire par une option sans fil.

Cependant, remplacer un bus filaire par une liaison sans fil dans une application aussi

critique est un défi. La communication sans fil utilise un support naturellement instable et est exposée aux interférences externes. Une solution de réseau WBMS doit garantir le niveau de sécurité requis par le VE, en assurant une fiabilité élevée, une latence limitée et en maintenant aussi faible que possible la consommation d'énergie supplémentaire par rapport à la version filaire. Étant donné que le WBMS a des exigences très particulières, la solution réseau doit être soigneusement conçue. Parmi les multiples options de réseau sans fil disponibles, aucune n'a été spécialement conçue pour cette application. Néanmoins, certains d'entre eux peuvent être adaptés ou combinés pour répondre à toutes les exigences. La norme IEEE. 802.15.4 est le document de référence pour les réseaux personnels sans fil à faible débit (LR-WPAN), qui a évolué au fil des années, améliorant ses performances et sa robustesse pour atteindre plusieurs cas d'utilisation. Une autre option serait le Bluetooth Low Energy (BLE), largement utilisé pour de multiples applications. Ces options utilisent des mécanismes de division temporelle et de diversité de fréquence pour obtenir une communication à faible puissance et très fiable. Ainsi, il y a une discussion ouverte sur le réseau sans fil le plus approprié pour le WBMS. Dans cette thèse, nous prétendons rejoindre le débat en présentant tous les aspects pertinents du problème, en proposant une solution basée sur l'état de l'art et en démontrant sa faisabilité avec des tests en conditions réelles.

A.1.1 Contribution

Ce manuscrit vise à discuter des défis liés à l'adoption du WBMS dans l'industrie automobile. Il fournit des informations pour résoudre le problème d'implémentation tout en gardant la robustesse nécessaire d'un système aussi critique. Notre objectif est de proposer, mettre en œuvre et tester une solution WBMS pour évaluer la convenance d'introduire ce nouveau système dans les véhicules électriques. Les contributions contenues dans ce document sont les suivantes :

— **Contexte et état de l'art:** Le travail de cette thèse rassemble deux sujets très différents ; VE et réseaux sans fil à faible consommation. Nous présentons les principaux enjeux du véhicule électrique, des batteries électriques et du BMS. Nous détaillons les exigences de sécurité et les risques liés à une mauvaise manipulation de ce genre de systèmes. Nous expliquons également le principe de fonctionnement des principaux candidats de protocole sans fil pour implémenter un réseau WBMS. De plus, nous présentons les options disponibles sur le marché liées au WBMS.

— **Communication sans fil robuste pour les applications critiques:** nous pro-

posons un modèle de réseau qui peut être adapté à plusieurs applications WSN nécessitant une haute fiabilité, une latence limitée et une faible consommation d'énergie, comme le WBMS. Notre proposition est basée sur le mode MAC TSCH de la norme IEEE Std. 802.15.4 avec une fonction d'ordonnancement personnalisé qui utilise la méthode d'accusé de réception de groupe (GACK) pour permettre la gestion dynamique de la retransmission. Le réseau proposé utilise la couche physique du BLE et une méthode de liste noire des canaux pour garantir des performances élevées même dans des environnements à fortes interférences externes. Nous présentons également certaines considérations matérielles et logicielles à prendre en compte lors de la mise en œuvre de notre WSN sur différentes plateformes.

— **Mode endormi pour WBMS:** Nous présentons un mode basse consommation pour notre réseau proposé, qui répond à toutes les exigences du WBMS. Notre approche consiste à passer d'une topologie en étoile centralisée en mode actif à une architecture distribuée en mode veille. Dans notre mode endormi, les nœuds peuvent maintenir la synchronisation et l'équilibre de la batterie sans aucune intervention du maître WBMS, ce qui évite la décharge indésirable de la batterie 12V du VE.

— **Révolution de l'architecture logicielle BMS:** L'introduction d'un réseau WBMS signifie qu'il est nécessaire d'ajouter un microcontrôleur à chaque esclave pour gérer l'échange de données. Cela signifie que nous avons une puissance de calcul supplémentaire par rapport à la version filaire, ce qui est l'occasion de modifier notre façon d'implémenter le logiciel BMS. Dans cette thèse, nous proposons et testons quelques nouvelles fonctionnalités qui tirent parti de cette nouvelle architecture distribuée et de la synchronisation donnée par le réseau sans fil.

— **Test de véhicule WBMS:** Nous présentons une étude de la compatibilité du WBMS avec les autres systèmes RF d'un VE. Nous mesurons également les effets des interférences externes sur la stabilité de la liaison du réseau lorsque le pack batterie est à l'intérieur du véhicule. Enfin, nous comparons les performances du réseau lorsque le véhicule est en mode conduite et en mode stationnement.

A.2 État de l'art

A.2.1 Véhicule électrique

Les véhicules électriques n'ont pas autant d'arrangements mécaniques complexes que les véhicules thermiques. Le moteur est la seule pièce mobile, reliée par des fils électriques à une source d'alimentation, généralement une batterie à courant continu (DC). Étant donné que les deux pièces n'ont pas besoin d'être côte à côte, les véhicules électriques sont flexibles et peuvent avoir plusieurs configurations possibles en fonction de l'application [11]. La Figure 2.1 présente une vue d'ensemble de l'architecture d'un VE [12]. Les batteries DC haute tension (HV) sont le moyen le plus simple de stocker de l'énergie. La batterie HV comprend un groupe de cellules et un BMS. Le moteur est chargé de convertir l'énergie électrique en énergie mécanique et inversement. L'idée principale est de faire tourner le rotor à l'aide d'un champ magnétique rotatif (MF) généré dans le stator par un courant alternatif (AC) [11]. Par conséquent, un onduleur DC-AC triphasé est nécessaire. Les véhicules électriques peuvent avoir plusieurs configurations. La Nissan Leaf ou la Renault Zoe a un moteur monté à l'avant. La Tesla Model S a une version à moteur unique montée à l'arrière et une autre à deux moteurs assurant une transmission intégrale [11].

Presque tous les véhicules électriques utilisent des cellules de batterie au lithium-ion (LIB) avec différentes configurations de matériaux. Le lithium (Li) est une excellente option pour les batteries car il est largement disponible, non toxique et très susceptible de perdre des électrons [19]. Malgré ses avantages significatifs, le lithium est hautement réactif, ce qui rend difficile la construction d'un système sûr avec cet élément. Le concept de fenêtre de sécurité, illustré à la Figure 2.5, est un espace bidimensionnel de valeurs optimales de température et de tension où une cellule LIB doit rester. Sortir des limites de cette fenêtre peut entraîner une dégradation permanente des performances des cellules de la batterie, voire des résultats catastrophiques dans le pire des cas [19]. Le BMS surveille la batterie et assure son bon fonctionnement. Il s'agit d'un groupe de capteurs de tension, de température et de courant qui collectent périodiquement des données pour estimer l'état de charge (SOC) et l'état de santé (SOH) des cellules. Sur la base de ces calculs, le BMS peut autoriser ou interdire la charge et la décharge de la batterie, définir une puissance de sortie et d'entrée maximale, alerter le dépassement des limites de la fenêtre de sécurité ou mettre en place le processus d'équilibrage pour égaliser la charge de toutes les cellules [23]. Un BMS a deux modes de fonctionnement, le mode veille qui est un état de faible puissance utilisé lorsque le véhicule est éteint et le mode actif utilisé lors de la

conduite. Le BMS utilise un topologie distribué où les capteur et l'unité de calcul ne sont pas dans la même carte électronique. Ces dispositifs utilisent une hiérarchie maître-esclave où le maître envoi des commandes aux esclaves sur un interface de communication filaire et attend la réponse sur le même lien. La Figure 2.7b montre un représentation du BMS avec liaison filaire sur un pack batterie.

A.2.2 Réseaux sans fil à faible consommation

En 2003, la norme IEEE 802.15.4 a été publiée en tant que document de référence pour les réseaux personnels sans fil à faible débit (LR-WPAN). Pour répondre aux exigences spécifiques de qualité de service (QoS), la norme a dû évoluer, c'est pourquoi en 2012, l'amendement IEEE 802.15.4e a été publié. L'amélioration significative comprend de nouveaux comportements de contrôle d'accès au support (MAC) pour obtenir une communication déterministe, une faible consommation d'énergie et une robustesse contre les interférences de fréquence [40].

Le schéma LLDN MAC apparaît pour les applications où il est nécessaire de récupérer les données du capteur au maximum toutes les 10 ms. Il ne propose qu'une topologie de réseau en étoile où un appareil central échange des données directement avec les périphériques. LLDN s'appuie sur le mécanisme d'accès multiple par répartition dans le temps (TDMA) pour atteindre une fiabilité élevée et une faible consommation d'énergie. LLDN propose d'utiliser le mécanisme d'accusé de réception de groupe (GACK) pour synchroniser la distribution des intervalles de temps de retransmission. Avec cette méthode, la racine n'a pas besoin d'envoyer un ACK individuel chaque fois qu'elle reçoit un paquet d'un nœud. Au contraire, il utilise l'intervalle de temps réservé GACK pour envoyer un message de diffusion indiquant quels paquets ont été correctement reçus. Sur la base de ce GACK, chaque nœud sait s'il peut s'endormir ou s'il doit retransmettre à un intervalle de temps spécifique [41]. Time Slotted Channel Hopping (TSCH) est un autre mode MAC introduit par l'amendement IEEE 802.15.4 de 2012, prenant en charge les topologies en étoile et maillées. Cette méthode utilise TDMA et l'accès multiple par répartition en fréquence (FDMA) pour obtenir une communication sans fil à haute fiabilité et à faible puissance [51]. Dans TSCH, le réseau n'utilisera pas le même canal de fréquence pour deux intervalles de temps consécutifs (saut de canal) [51], comme vous pouvez le voir sur la Figure 2.14. L'utilisation de canaux différents pour chaque intervalle de temps rend le réseau plus robuste aux problèmes d'interférence et d'entrelacement de chemin. Contrairement à LLDN, pour TSCH, la norme ne propose pas comment le réseau

doit planifier l'échange de données entre les nœuds. Ils préfèrent laisser cette tâche aux couches supérieures.

Les réseaux BLE utilisent la division temporelle et la diversité de fréquence avec une modulation GFSK à 2 Mb/s. Ces réseaux présentent une grande fiabilité car ils exécutent la retransmission nécessaire jusqu'à la livraison réussie, mais cela ne garantit pas une latence spécifique. Nous avons également examiné Wireless IO-Link, qui fournit une latence limitée pour les petits réseaux. Chaque maître prend en charge quatre esclaves avec des charges utiles de 25 octets ou jusqu'à 8 esclaves avec une longueur de message plus petite.

Après cette revue des protocoles, nous pouvons constater que même si ces réseaux présentent plusieurs avantages, aucun d'entre eux ne peut garantir toutes les exigences particulières d'une application WBMS. Mais nous avons remarqué que la division temporelle, la diversité de fréquence, la liste noire des canaux et une bonne gestion de la retransmission sont les clés pour créer un réseau adapté pour remplacer une interface filaire dans un système critique comme BMS.

A.3 Communication sans fil robuste pour WBMS

A.3.1 Communication sans fil pour les systèmes critiques

Afin de permettre l'inclusion de la communication sans fil dans les applications WSN critiques, nous avons développé un modèle de réseau qui peut être adapté à de multiples applications nécessitant une haute fiabilité, une latence limitée et une faible consommation d'énergie. Dans ce travail, nous proposons un réseau TSCH utilisant un ordonnancement centralisé basé sur GACK avec un algorithme de liste noire de canaux. Notre réseau utilise une topologie en étoile fonctionnant sur la couche physique BLE. La topologie en étoile garantit une consommation d'énergie similaire des nœuds, tandis que la méthode GACK permet une gestion dynamique de la retransmission. Le protocole TSCH MAC tire parti de la division temporelle pour éviter les collisions et le saut de canal radio pour atténuer les interférences externes. La couche physique BLE nous permet d'atteindre un débit plus élevé et une durée de timeslot plus courte. La Figure 3.2 présente notre proposition de fonction d'ordonnancement pour un WSN composé d'un dispositif central et de N nœuds où la racine doit recevoir un nouveau paquet des périphériques périodiquement, tous le TRT millisecondes. La racine est chargée d'attribuer un identifiant unique (ID de 1 à N)

au sein du réseau à chaque nœud lors de la phase d'initialisation.

Notre fonction d'ordonnancement est divisée en deux parties, la première pour la transmission statique et la seconde pour la retransmission dynamique. Le premier intervalle de temps est réservé à la trame balise pour maintenir la synchronisation et permettre à de nouveaux nœuds d'identifier et de rejoindre le réseau. Ensuite, nous avons N intervalles de temps de liaison montante pour chaque nœud esclave où l'accusé de réception individuel n'est pas requis car nous utilisons le mécanisme GACK. Ces intervalles de temps sont distribués en fonction de l'ID de nœud, de sorte que le nœud avec l'ID égal à 1 utilise le premier intervalle de temps de liaison montante. Étant donné que la communication sans fil est naturellement instable, le nombre de retransmissions requises dans chaque tranche de temps est variable. Pour gérer efficacement la distribution des intervalles de temps de retransmission, nous utilisons deux trames GACK pour informer les esclaves des messages qui n'ont pas été reçus par la racine. Nous avons choisi cette approche pour minimiser la probabilité de perdre une trame GACK, car la racine peut l'envoyer deux fois sur différents canaux radio. Les informations GACK sont envoyées dans un format bitmap où chaque bit est dédié à informer un nœud spécifique en fonction de son ID. Si le bit correspondant est mis à 1, le message est perdu et une retransmission est nécessaire (acquittement négatif). Si le message a été livré avec succès, le nœud passe en mode "veille". Sur la base de la trame GACK, les nœuds peuvent déterminer l'intervalle de temps de retransmission qu'ils peuvent utiliser dans la deuxième partie de l'ordonnancement pour éviter les collisions de messages au sein du réseau.

A.3.2 WBMS pour le Renault Zoe

WBMS est l'application cible pour notre modèle de réseau basé sur TSCH avec retransmission dynamique. Nous avons implémenté ce réseau sans fil robuste pour le mode actif du WBMS d'une Renault Zoe afin de valider notre proposition décrite dans la section précédente. Le pack batterie de la Renault Zoe est composé de 12 modules contenant chacun 8 cellules. Ainsi, notre réseau devrait être composé d'une racine, le maître WBMS, et de 12 nœuds, les esclaves WBMS. Le matériel sélectionné pour la racine est le microcontrôleur Texas Instrument (TI) cc26x2. Pour les nœuds, nous utilisons une carte personnalisée (esclave WBMS) fournie par TI qui peut être branchée à l'interface des modules de batterie. Cette carte est principalement composée d'un microcontrôleur TI CC2642 pour la gestion du réseau et d'un ASIC BMS (TI BQ79616) en charge des mesures des cellules de la batterie, comme représenté sur la Figure 3.7a. Nous validons notre solution

en menant une série d'expériences dans le monde réel, montrant que notre proposition surpasse un protocole de retransmission statique traditionnel et peut atteindre 100 % de fiabilité de bout en bout avec une latence limitée et une faible consommation d'énergie (400 uA en moyenne par esclave) . Enfin, nous avons testé notre réseau sous forte interférence externe, où nous avons montré l'importance du mécanisme de liste noire dans ce genre d'environnement difficile. Notre campagne expérimentale démontre que le mécanisme GACK, en conjonction avec le mode TSCH MAC, rend possible la communication sans fil sur des applications critiques telles que le WBMS sans compromettre le niveau de sécurité du système global.

A.4 Mode endormi pour WBMS

Le BMS a deux modes de fonctionnement, le mode actif et le mode veille. Lorsque le véhicule électrique roule ou se charge, le BMS collecte périodiquement les données des cellules de la batterie (par exemple, toutes les 100 ms), comme décrit dans la section précédente. Lorsque le véhicule est garé, le BMS passe en mode veille, où il n'a qu'à se réveiller (par exemple, toutes les 8 heures) pour vérifier l'équilibre de charge des cellules de la batterie. Un BMS doit pouvoir passer du mode veille au mode actif dans un délai précis (1 seconde pour la Renault Zoe). De plus, la consommation d'énergie dans ce mode devrait être négligeable, soit moins de 70 µA par esclave pour la Renault Zoe. Dans un BMS filaire, éteindre les esclaves en mode veille est facile car nous pouvons les réveiller grâce à une interruption envoyée sur le bus UART. Pour un WBMS utilisant un réseau synchronisé, nous ne pouvons pas simplement éteindre complètement les esclaves car nous n'avons aucun lien physique pour envoyer une interruption de réveil. En fait, il faut maintenir les nœuds synchronisés ; sinon, les esclaves doivent garder la radio allumée pendant le mode veille en écoutant le médium jusqu'à ce qu'ils reçoivent l'ordre de réveil, ce qui consommerait tellement d'énergie (8 mA en moyenne pour le microcontrôleur TI CC26x2). Le moyen le plus simple d'implémenter le mode veille pour un WBMS consiste à créer un nouveau ordonnancement dans lequel le maître diffuse périodiquement un message de balise (EB) pour maintenir la synchronisation du réseau. L'inconvénient de cette méthodologie est que puisque le maître est alimenté par la petite batterie 12V de la voiture qui n'a pas une capacité si importante comme la pack batterie, il existe le risque de la décharger si le maître ne s'endort pas complètement. Nous avons conçu une alternative pour garder la synchronisation entre les nœuds sans utiliser le maître en mode veille, tout

cela sans dépasser le temps de réveil ou les limites de consommation d'énergie.

Nous proposons de ne pas choisir un nœud central unique pour synchroniser le réseau mais d'utiliser tous les nœuds avec un horaire prédéfini. Pour conserver la synchronisation, tous les T_{sync}, un nœud différend doit diffuser un EB pour réduire la dérive d'horloge entre les esclaves. Cela signifie que chaque nœud allume sa radio à chaque T_{sync} pour recevoir un EB d'un autre nœud ou pour envoyer un EB quand son tour arrive. Avec cette méthode, chaque nœud a la même charge de travail, et par conséquent, tous ont la même consommation d'énergie. De plus, chaque nœud peut inclure les mesures de tension de son module de batterie dans le message EB. De cette façon, chaque nœud reçoit toutes les tensions des cellules du pack batterie et peut décider quelles cellules de son module doivent être déchargées pour maintenir l'équilibre. Pour passer du mode veille au mode actif, le maître WBMS doit se resynchroniser sur le réseau et redevenir la source horaire des esclaves. Lorsque le maître reçoit la commande de réveil de l'ordinateur central du VE via une interruption, il doit allumer sa radio et écouter un EB d'un esclave WBMS. Dès qu'il reçoit le message de synchronisation, il peut commencer à envoyer des EB, indiquant aux esclaves de passer en mode actif.

Le mode endormi proposé pour le WBMS a été implémenté pour la batterie de la Renault Zoe. Nous utilisons le même matériel présenté dans la section précédente avec la carte personnalisée pour les esclaves WBMS. Le réseau a été testé en mode veille pendant 3 jours et les nœuds n'ont jamais perdu la synchronisation. Nous avons également mesuré la consommation énergétique des esclaves en utilisant l'outil TI Energy Trace. Les résultats montrent un courant moyen de 60 µA par esclave. En plus, nous présentons un mode veille WBMS amélioré où nous augmentons la période de synchronisation des esclaves sans affecter les propriétés de temps de réveil de la méthode précédente. Cette nouvelle version offre une réduction de 20 % de la consommation par esclave.

A.5 Au-delà du WBMS

Le WBMS nécessite l'inclusion d'un microcontrôleur RF dans chaque esclave, qui dispose d'une interface physique avec l'ASIC BMS et supervise la gestion des communications sans fil. Ainsi, dans un WBMS, on gagne une capacité de calcul distribuée par rapport à un BMS traditionnel. Même si ces microcontrôleurs n'ont pas de microprocesseurs super puissants, ils peuvent toujours effectuer certaines opérations pour réduire la charge de travail de l'appareil principal. L'introduction d'un WBMS est l'occasion de modifier

le fonctionnement du logiciel BMS afin d'améliorer certaines caractéristiques du système et de réduire les coûts. Comme cela permet d'effectuer certains calculs en parallèle, le dispositif central n'a pas besoin d'être aussi puissant que dans le BMS traditionnel. Un exemple serait la mesure de la tension de la batterie. Dans notre implémentation WBMS, nous avons une application logicielle dans les nœuds qui envoie périodiquement une commande à l'ASIC BMS pour mesurer la tension de la cellule, attend la réponse, calcule la moyenne et transmet les données au maître WBMS. Tout cela sans intervention du CPU central. Un autre exemple serait la mesure de température, dans un BMS traditionnel l'unité centrale ne reçoit pas la mesure directe mais une tension GPIO qu'il faut traiter pour obtenir la valeur de température. Notre application WBMS dans les esclaves exécute ces calculs avant d'envoyer les données, ce qui réduit la charge de travail de l'unité centrale.

Nous proposons d'utiliser cette capacité de calcul des esclaves pour compenser l'offset de tension produit par l'équilibrage de la cellule afin d'augmenter ses performances. Lorsque les cellules du pack batterie n'ont pas le même niveau de charge, le BMS doit décharger les cellules de la batterie avec le niveau le plus élevé jusqu'à ce que toutes les cellules aient la même charge. En règle générale, les ASIC BMS sont préparés pour cette opération, car ils incluent des FET qui activent/désactivent la décharge des cellules. Le principal inconvénient de cette fonctionnalité est qu'elle partage une partie de son circuit avec la fonction de mesure, de sorte que la décharge de la cellule produit une perturbation dans la mesure de la tension. Dans une application typique, lorsque le BMS doit mesurer les tensions des cellules pendant le processus d'équilibrage des cellules, il interrompt le mécanisme de décharge pour éviter ces perturbations. Dans cette approche, le temps nécessaire pour effectuer une nouvelle mesure dépend du filtre RC utilisé dans l'esclave. En utilisant cette approche, dans notre solution WBMS, l'esclave doit mettre en pause l'équilibrage de la cellule pendant 1,5 ms (temps de stabilisation + temps de mesure) à chaque fois qu'il a besoin d'une nouvelle mesure. Suspendre l'équilibrage des cellules pendant 1,5 ms toutes les 20 ms conduirait à perdre 7,5 % du temps de décharge effectif (75 ms par seconde). En 8 heures, cela signifierait 36 minutes perdues. Pour réduire les intervalles de pause d'équilibrage, nous proposons un algorithme pour compenser l'offset de tension causé par l'opération de décharge. Notre algorithme proposé a été implémenté dans notre solution WBMS et renvoie des valeurs dans un intervalle de 2 mV , ce qui est très proche des performances de l'approche classique (1 mV) avec un temps d'attente de 1,5 ms. Le principal avantage est que le système n'a besoin de suspendre l'équilibrage

que pendant 3 ms par seconde (0,3 %) avec notre proposition. De plus, avec la capacité distribuée WBMS, il n'y a pas d'intervention nécessaire du maître pour exécuter cette tâche.

L'estimation SOC est très utile pour comprendre l'état de la batterie. En règle générale, un BMS modélise le pack batterie comme une seule cellule et calcule un seul SOC pour cette cellule moyenne. L'équipe Renault BMS préfère estimer le SOC de chaque cellule et le stocker dans le temps, car il est utile d'obtenir une bonne estimation de la santé des cellules de la batterie. Estimer périodiquement tout le SOC de la batterie (par exemple, 96 cellules toutes les 100 ms) dans un seul microprocesseur central nécessite une puissance de calcul considérable, ce qui augmenterait les coûts du BMS et sa consommation d'énergie. Nous avons présenté le processus d'inclusion d'un algorithme de filtrage de Kalman dans les microcontrôleurs RF de basse consommation. Nous avons modifié la fonction d'ordonnancement pour donner le temps libre nécessaire aux microprocesseurs pour exécuter la mise à jour du SOC à chaque slotframe. Dans nos tests, il ne faut que 6 ms à chaque nœud pour calculer une nouvelle valeur pour l'ensemble des 8 cellules de son module. Cela peut être considéré comme la première étape d'une approche "module intelligent", où un module de batterie comprendra un esclave WBMS pendant sa durée de vie, qui peut stocker l'historique d'utilisation des cellules et estimer le SOC ou le SOH sans l'intervention d'une unité centrale.

A.6 Essais sur véhicule

Tout au long de ce document, nous avons décrit une solution complète pour mettre en œuvre un WBMS robuste. Jusqu'à présent, tous les mécanismes proposés étaient testés en laboratoire. Il est temps maintenant de mettre notre système à l'intérieur d'un véhicule électrique pour démontrer ses performances dans des environnements réels. La première étape a consisté à adapter la racine WBMS avec une interface CAN pour interagir avec elle et recevoir toutes les données collectées sur le réseau. Nous avons également ajouté une capacité FOTA à notre système, ce qui nous permet de mettre à jour le micrologiciel des nœuds sans avoir à démonter la batterie du véhicule. Une fois le système prêt, nous avons installé la batterie dans une Renault Zoe. Lors de la première campagne de test, nous avons placé le maître du réseau dans différentes positions à l'extérieur du bloc-batterie pour voir s'il était possible de communiquer avec les nœuds. Les résultats montrent que c'est possible selon l'endroit où l'on place la racine. En général, lorsque le maître est à

plus de 40cm du véhicule, la racine n'arrive pas à rejoindre le réseau. Comme prévu, la force du signal était beaucoup plus faible par rapport au cas où le nœud central était placé à l'intérieur de la batterie. Ces résultats montrent qu'il serait possible de connecter un agent externe au réseau extérieur au véhicule pour obtenir l'état du réseau, mettre à jour le firmware, ou encore diagnostiquer le système en cas de panne. Notez que dans nos tests, nous utilisons un master basé sur une carte TI cc26x2 Launchpad avec une antenne imprimée. La qualité de la connexion peut être améliorée si nous utilisons du matériel avec une configuration d'antenne différente avec une meilleure sensibilité. Cependant, ces résultats indiquent que même si le réseau est enfermé à l'intérieur d'une structure métallique, du point de vue de la sécurité, il devrait être conçu de la même manière que si la barrière n'existait pas.

Nous avons ensuite testé la stabilité du réseau en présence d'interférences externes générées par certains routeurs Wi-Fi et quatre ordinateurs utilisant le protocole SCP pour générer un trafic constant. Pour ces expériences, nous avons comparé différentes configurations pour comprendre l'impact de la position de l'émetteur-récepteur d'interférence et de l'enceinte métallique. Nous avons constaté que placer le couvercle de la batterie et mettre le pack à l'intérieur du véhicule réduit l'impact des interférences externes de 25 %. Nous avons également vu que si le couvercle de la batterie comporte des trous de refroidissement et que nous plaçons les émetteurs-récepteurs à l'intérieur de la cabine à proximité de ces ouvertures, la stabilité des canaux du réseau est fortement détériorée. Cela dépend de la conception de la batterie et doit être pris en compte si vous utilisez un WBMS. Ces résultats ont montré que l'enceinte métallique ne protège pas entièrement le réseau des interférences externes, donc l'utilisation d'une méthode de liste noire des canaux est nécessaire pour maintenir la haute fiabilité du système.

A.7 Conclusion

L'industrie automobile est en pleine transition. Pendant plus de 100 ans, ils se sont consacrés à la production de véhicules thermiques (ICE). Aujourd'hui, d'un point de vue environnemental et juridique, la production de véhicules ICE doit s'arrêter et les véhicules électriques sont considérés comme le meilleur remplacement. L'élément le plus critique d'un VE est la batterie. Une batterie de VE est composée de cellules regroupées en modules ; par exemple, la batterie Renault Zoe a 96 cellules et 12 modules. Afin de garantir le niveau de sécurité requis, chaque VE utilise un BMS. Cette thèse vise à

proposer une solution pour remplacer l'interface filaire à l'intérieur du BMS par une liaison sans fil. Cela peut apporter plusieurs avantages, comme une conception plus simple du bloc-batterie, un gain d'espace, une réduction des risques de défaillance des connecteurs ou la possibilité d'un assemblage entièrement automatisé du bloc-batterie.

Dans le chapitre 2, nous avons présenté un aperçu des différentes options de protocoles sans fil pour implémenter le réseau d'un WBMS. Nous avons observé que même si ces réseaux présentent plusieurs avantages, aucun d'entre eux n'a été spécialement conçu pour une application telle que le WBMS. Cependant, nous avons vu que des mécanismes tels que la division temporelle, la diversité de fréquence, la retransmission GACK et la liste noire des canaux avaient montré d'excellents résultats dans l'amélioration des performances des réseaux sans fil à faible puissance. Dans la suite du document, nous avons présenté et testé notre proposition WBMS. Pour le mode actif, nous avons proposé un modèle de réseau pour les applications critiques avec une fiabilité élevée, une latence limitée et une faible consommation d'énergie. Nous avons conçu un réseau basé sur TSCH fonctionnant sur la couche physique BLE. En plus de cela, nous avons développé une fonction d'ordonnancement qui utilise une approche similaire à la méthode GACK du mode MAC LLDN pour gérer dynamiquement les retransmissions. De plus, nous avons créé un algorithme de liste noire des canaux adapté à notre fonction de planification pour maintenir la robustesse du réseau même dans des environnements très perturbés. Nous implémentons notre WBMS dans un matériau TI spécialement conçu pour le pack batterie Renault Zoe afin de valider notre proposition. Après plusieurs tests, nous avons sondé que notre méthode surpasse une approche de retransmission traditionnelle ; il peut atteindre 100 % de fiabilité même lorsque le réseau est soumis à de fortes interférences externes tout en maintenant la latence dans la limite requise et avec une consommation de courant moyenne de 400 µA par esclave.

Pour le mode veille, nous avons proposé de modifier l'architecture du réseau, où les nœuds échangent des balises entre eux et maintiennent la synchronisation sans avoir besoin d'avoir la racine WBMS éveillée. De cette façon, nous pouvons empêcher le nœud central de décharger la petite batterie 12 V du véhicule. De plus, notre mode veille proposé atteint 40 µA de consommation de courant moyenne par esclave et peut être utilisé pour maintenir l'équilibre de la batterie sans intervention de l'unité centrale. Dans le chapitre 5, nous avons discuté des nouvelles opportunités et des changements pour le logiciel BMS qu'apporte l'introduction d'un réseau sans fil. Nous proposons d'utiliser la capacité de calcul des esclaves WBMS pour compenser l'offset de tension produit par le processus

d'équilibrage des cellules ou même pour calculer le SOC des cellules. Nos résultats prouvent que la nouvelle capacité de calcul distribuée du WBMS peut réduire la charge de travail du BMS central et influencer de manière significative les possibilités d'applications de seconde vie pour les modules de batterie. Enfin, nous avons mis notre système dans une Renault Zoe et effectué plusieurs tests pour montrer les performances de notre proposition dans un environnement réel.

Le système conçu a répondu aux exigences d'une telle application critique et a été testé dans plusieurs scénarios, montrant la viabilité du WBMS. L'adoption de cette application sans fil est une réalité car de nombreux fabricants de véhicules électriques ont décidé de l'adopter pour leurs prochaines versions. Cependant, certains sujets ouverts méritent d'être mentionnés pour utiliser le potentiel de cette nouvelle architecture prometteuse pour le BMS. Nos tests ont montré que l'introduction du réseau sans fil à l'intérieur de la batterie pouvait entraîner un diagnostic et une réparation du système BMS sans démonter la batterie du véhicule électrique. Cette possibilité peut économiser beaucoup de temps et d'efforts dans la vie après-vente d'un VE équipé d'un WBMS. Cependant, cela ouvre également la porte à des risques de sécurité potentiels pour le système car les signaux du WBMS sont exposés au monde extérieur. Nous avons également sondé qu'il serait possible d'utiliser les propriétés des ondes électromagnétiques pour intégrer le maître WBMS au microcontrôleur central du VE, qui est généralement placé à l'extérieur de la batterie. Notre expérience dans la Renault Zoe indique que nous pouvons atteindre le réseau depuis l'extérieur du pack batterie en générant les chemins appropriés pour les signaux électromagnétiques. L'ajout des fonctions centrales du BMS à un autre microcontrôleur de véhicule réduirait la quantité de matériel nécessaire dans le véhicule électrique. Cependant, cette fonctionnalité nécessiterait une étude supplémentaire de la dispersion du signal électromagnétique du véhicule, ce qui peut être coûteux et rendre la solution potentiellement non viable.

Les microcontrôleurs esclaves WBMS peuvent tracer les conditions historiques des modules de batterie et surveiller les cellules tout au long de leur durée de vie. Idéalement, dans les applications de seconde vie, nous devrions pouvoir utiliser le même matériel et les mêmes données pour optimiser l'utilisation des cellules de la batterie. La possibilité de modifier le logiciel des nœuds par liaison radio ajouterait une flexibilité supplémentaire pour utiliser les esclaves dans plusieurs applications. Néanmoins, pour tirer le meilleur parti de ces avantages du WBMS, il est nécessaire d'avoir un certain niveau de standardisation qui n'existe pas actuellement.

La sûreté est une préoccupation centrale dans le monde automobile. Le BMS filaire de Renault Zoe met en œuvre le niveau de sûreté le plus élevé de l'industrie (ASIL D). La mise en œuvre d'un WBMS compatible ASIL D est possible, mais ce n'est pas une tâche facile. Texas Instrument utilise le concept de canal noir, où ils utilisent le microcontrôleur RF et le réseau comme support de transmission qui ne modifie pas l'échange de données entre le maître WBMS et les ASIC BMS. Cette approche est valide mais rend impossible l'utilisation de la capacité de calcul supplémentaire des nœuds comme nous l'avons proposé dans cette thèse. Une méthode pour réaliser un système ASIL D et exécuter des calculs dans les esclaves consisterait à ajouter un deuxième microprocesseur pour chaque nœud qui surveille le comportement du microcontrôleur RF principal. Cette solution ajouterait un coût supplémentaire qui pourrait rendre l'introduction d'une solution WBMS économiquement non viable. Les travaux futurs sur le WBMS pourraient se concentrer sur l'étude de différentes architectures matérielles et logicielles pour atteindre le niveau de sûreté maximal de l'industrie automobile tout en gardant le WBMS attractif d'un point de vue économique.

PUBLICATIONS

Conference Proceedings

— F. A. Rincon Vija, S. Cregut, G. Z. Papadopoulos and N. Montavont, "Enabling Robust Wireless Communication for BMS on Electric Vehicles," 2021 IEEE 46th Conference on Local Computer Networks (LCN), 2021, pp. 423-426, Available: www.ieeexplore.ieee.org/document/9524937

— F. A. Rincon Vija, S. Cregut, G. Z. Papadopoulos and N. Montavont, "From Wired to Wireless BMS in Electric Vehicles," 2021 17th International Conference on Mobility, Sensing and Networking (MSN), 2021, pp. 255-262, Available: www.ieeexplore.ieee.org/document/9751479

— F. A. Rincon Vija, S. Cregut, G. Z. Papadopoulos and N. Montavont, "Autonomous Balancing Sleep Mode for Wireless BMS in Electric Vehicles," 2021 IEEE Conference on Standards for Communications and Networking (CSCN), 2021, pp. 65-71, Available: www.ieeexplore.ieee.org/document/9686103

Demonstrations in International Conferences

— F. A. Rincon Vija, S. Cregut, G. Z. Papadopoulos and N. Montavont, "Demo: Wireless Battery Management System", EWSN 2022-ACM International Conference on Embedded Wireless Systems and Networks. 2022.

List of Figures

List of Tables

Acronyms

AC	Alternating Current.
ADI	Analog Devices.
Ah	Ampere Hour.
ASIC	Application-Specific Integrated Circuit.
ASIL	Automotive Safety Integrity Level.
ASN	Absolute Slot Number.
BBR	Bluetooth Basic Rate.
BLE	Bluetooth Low Energy.
BMS	Battery Management System.
DC	Direct Current.
DODAG	Destination Oriented Directed Acyclic Graph.
DSSS	Direct Sequence Spread Spectrum.
ECU	Electronic Control Unit.
EESM	Externally Excited Synchronous Motor.
EIS	Electrochemical Impedance Spectroscopy.
EKF	Extended Kalman Filtering.
EMI	Electromagnetic Interference.
EV	Electric Vehicles.
FDMA	Frequency Division Multiple Access.
FSK	Frequency Shift Keying.
GACK	Group Acknowledgement.
GFSK	Gaussian Frequency Shift Keying.
HARA	Hazard Analysis and Risk Assessment.

HV	High Voltage.
ICE	Internal Combustion Engine.
IETF	Internet Engineering Task Force.
IOLW	IO-Link wireless.
ISM	Industrial, Scientific and Medical.
LIB	Lithium-ion-based Battery.
LLDN	Low Latency Deterministic Networks.
LR-WPAN	Low-Rate Wireless Personal Area Networks.
LV	Low Voltage.
MAC	Medium Access Control.
MF	Magnetic Field.
OCV	Open Circuit Voltage.
OQPSK	Offset Quadrature Phase-Shift Keying.
PDR	Packet Delivery Ratio.
PHY	Physical.
PM	Permanent Magnet.
PMSM	Permanent Magnet Synchronous Motor.
QoS	Quality of Service.
QPSK	Quadrature Phase-Shift Keying.
RSSI	Received Signal Strength Indication.
SNR	Signal to Noise Ratio.
SOC	State of Charge.
SOH	State of Health.
SPI	Serial Peripheral Interface.
TDMA	Time Division Multiple Access.

TI	Texas Instrument.
TRT	Target Refresh Time.
TSCH	Time Slotted Channel Hopping.
UART	Universal Asynchronous Receiver Transmitter.
WBMS	Wireless Battery Management System.
WMEWMA	Window Mean Exponential Weighted Moving Average.
WSN	Wireless Sensor Networks.
WuR	Wake-up Radio.

Bibliography

[1] James Larminie and John Lowry, *Electric vehicle technology explained*, John Wiley & Sons, 2012.

[2] Guillaume Le Gall, « Wireless network for reliable electric vehicle battery management », PhD thesis, Ecole nationale supérieure Mines-Télécom Atlantique, 2021.

[3] Avere France, *L'histoire du véhicule électrique*, Aug. 2022, URL: `www.avere-france.org/lhistoire-du-vehicule-electrique/` (visited on 08/07/2022).

[4] Eduardo Valsera-Naranjo et al., « Electrical vehicles: State of art and issues for their connection to the network », *in*: *2009 10th International Conference on Electrical Power Quality and Utilisation*, IEEE, pp. 1–3.

[5] European-Commission, *Paris Agreement*, Sept. 2022, URL: `ec.europa.eu/clima/eu-action/international-action-climate-change/climate-negotiations/paris-agreement_en` (visited on 09/07/2022).

[6] France24, *EU Parliament approves ban on new fossil-fuel cars from 2035*, June 2022, URL: `www.france24.com/en/europe/20220609-eu-parliament-approves-ban-on-new-fossil-fueled-cars-by-2035` (visited on 09/07/2022).

[7] Réseau de Transport d'Électricité (RTE), *Enjeux du développement de l'électromobilité pour le système électrique*, Report, May 2019, 80 pp.

[8] Bruno Scrosati, « History of lithium batteries », *in*: *Journal of solid state electrochemistry* 15.*7* (2011), pp. 1623–1630.

[9] Norbert Bieler and Paul Hartanto-Doeser, « wBMS Technology: The New Competitive Edge for EV Manufacturers », *in*: *Analog Devices* (2022).

[10] Wanbin Ren, Danyang Du, and Yang Du, « Electrical contact resistance of connector response to mechanical vibration environment », *in*: *IEEE Transactions on Components, Packaging and Manufacturing Technology* 10.*2* (2019), pp. 212–219.

[11] Fuad Un-Noor et al., « A comprehensive study of key electric vehicle (EV) components, technologies, challenges, impacts, and future direction of development », *in*: *Energies* 10.*8* (2017), p. 1217.

[12] David Marcos et al., « A safety concept for an automotive lithium-based battery management system », *in*: *2019 Electric Vehicles International Conference (EV)*, IEEE, pp. 1–6.

[13] Martin Lukasiewycz et al., « System architecture and software design for electric vehicles », *in*: *2013 50th Annual Design Automation Conference*, pp. 1–6.

[14] Emma Arfa Grunditz and Torbjörn Thiringer, « Performance analysis of current BEVs based on a comprehensive review of specifications », *in*: *IEEE Transactions on Transportation Electrification* 2.*3* (2016), pp. 270–289.

[16] Sang-Hwa Do et al., « Torque ripple reduction of wound rotor synchronous motor using rotor slits », *in*: *2012 15th International Conference on Electrical Machines and Systems (ICEMS)*, IEEE, pp. 1–4.

[17] Bertrand Revol et al., « EMI study of three-phase inverter-fed motor drives », *in*: *IEEE Transactions on Industry Applications* 47.*1* (2010), pp. 223–231.

[18] Kodjo Mawonou, « Développement d'algorithmes adaptatifs embarqués et débarqués du système de gestion batterie pour l'estimation des états de la batterie en usage automobile », PhD thesis, Université Paris-Saclay, 2020.

[19] Ghassan Zubi et al., « The lithium-ion battery: State of the art and future perspectives », *in*: *Renewable and Sustainable Energy Reviews* 89 (2018), pp. 292–308.

[20] Mahammad A Hannan et al., « State-of-the-art and energy management system of lithium-ion batteries in electric vehicle applications: Issues and recommendations », *in*: *Ieee Access* 6 (2018), pp. 19362–19378.

[21] Jianwu Wen, Yan Yu, and Chunhua Chen, « A review on lithium-ion batteries safety issues: existing problems and possible solutions », *in*: *Materials express* 2.*3* (2012), pp. 197–212.

[22] Dion Hubble et al., « Liquid electrolyte development for low-temperature lithium-ion batteries », *in*: *Energy & Environmental Science* (2022).

[23] Habiballah Rahimi-Eichi et al., « Battery management system: An overview of its application in the smart grid and electric vehicles », *in*: *IEEE industrial electronics magazine* 7.*2* (2013), pp. 4–16.

[24] Gregory L Plett, « Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 1. Background », *in*: *Journal of Power sources* 134.*2* (2004), pp. 252–261.

[25] Gregory L. Plett, « Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 2. Modeling and identification », *in*: *Journal of Power Sources* 134.*2* (2004), pp. 262–276, ISSN: 0378-7753, DOI: https://doi.org/10.1016/j.jpowsour.2004.02.032, URL: https://www.sciencedirect.com/science/article/pii/S037877530400360X.

[26] Ghufron Fathoni et al., « Comparison of State-of-Charge (SOC) estimation performance based on three popular methods: Coulomb counting, open circuit voltage, and Kalman filter », *in*: *2017 2nd International Conference on Automation, Cognitive Science, Optics, Micro Electro-Mechanical System, and Information Technology (ICACOMIT)*, IEEE, pp. 70–74.

[27] Yunqiu Wang et al., « Lithium-ion battery SOC estimation based on an improved adaptive extended Kalman filter », *in*: *2021 IEEE 16th Conference on Industrial Electronics and Applications (ICIEA)*, IEEE, pp. 417–421.

[28] Anthony Barré et al., « A review on lithium-ion battery ageing mechanisms and estimations for automotive applications », *in*: *Journal of Power Sources* 241 (2013), pp. 680–689.

[29] Jeffrey R Belt et al., « A capacity and power fade study of Li-ion cells during life cycle testing », *in*: *Journal of Power Sources* 123.*2* (2003), pp. 241–246.

[30] Ira Bloom et al., « An accelerated calendar and cycle life study of Li-ion cells », *in*: *Journal of power sources* 101.*2* (2001), pp. 238–247.

[31] Dian Wang, Yun Bao, and Jianjun Shi, « Online lithium-ion battery internal resistance measurement application in state-of-charge estimation using the extended kalman filter », *in*: *Energies* 10.*9* (2017), p. 1284.

[32] Manoj Mathew et al., « Comparative analysis of lithium-ion battery resistance estimation techniques for battery management systems », *in*: *Energies* 11.*6* (2018), p. 1490.

[34] Shweta Koraddi et al., « Analysis of Different Cell Balancing Techniques », *in*: *2022 International Conference for Advancement in Technology (ICONAT)*, IEEE, pp. 1–4.

[35] Seonwoo Jeon, Jae-Jung Yun, and Sungwoo Bae, « Active cell balancing circuit for series-connected battery cells », *in*: *2015 9th International Conference on Power Electronics and ECCE Asia (ICPE-ECCE Asia)*, IEEE, pp. 1182–1187.

[36] Geon-Hong Min and Jung-Ik Ha, « Active cell balancing algorithm for serially connected li-ion batteries based on power to energy ratio », *in*: *2017 IEEE Energy Conversion Congress and Exposition (ECCE)*, IEEE, pp. 2748–2753.

[37] Seo-Hyun Jeon et al., « Automotive hardware development according to ISO 26262 », *in*: *2011 13th international conference on advanced communication technology ICACT*, IEEE, pp. 588–592.

[38] Rick Salay, Rodrigo Queiroz, and Krzysztof Czarnecki, « An analysis of ISO 26262: Using machine learning safely in automotive software », *in*: *arXiv preprint arXiv: 1709.02435* (2017).

[39] Jon T Adams, « An Introduction to IEEE STD 802.15.4 », *in*: *2006 IEEE Aerospace Conference*, IEEE, 8–pp.

[40] Harrison Kurunathan et al., « IEEE 802.15. 4e in a Nutshell: Survey and Performance Evaluation », *in*: *IEEE Communications Surveys & Tutorials* 20.*3* (2018), pp. 1989–2010.

[41] Mashood Anwar and Yuanqing Xia, « IEEE 802.15. 4e LLDN: Superframe Configuration for Networked Control Systems », *in*: *2014 33rd Chinese Control Conference*, IEEE, pp. 5568–5573.

[42] « IEEE Standard for Local and Metropolitan Area Networks–Part 15.4: LR-WPANs », *in*: *IEEE Std 802.15.4e-2012 (Amendment to IEEE Std 802.15.4-2011)* (2012), pp. 1–225, DOI: 10.1109/IEEESTD.2012.6185525.

[43] « IEEE Standard for Low-Rate Wireless Networks », *in*: *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)* (2016), pp. 1–709, DOI: 10.1109/IEEESTD.2016.7460875.

[44] Luca Dariz, Giorgio Malaguti, and Massimiliano Ruggeri, « Performance Analysis of IEEE 802.15. 4 Real Time Enhancement », *in*: *2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE)*, IEEE, pp. 1475–1480.

[45] Gaetano Patti, Giuliana Alderisi, and Lucia Lo Bello, « Introducing Multi Level Communication in the IEEE 802.15. 4e Protocol: The MultiChannel-LLDN », *in*: *2014 IEEE Emerging Technology and Factory Automation (ETFA)*, IEEE, pp. 1–8.

[46] Gaetano Patti and Lucia Lo Bello, « A Priority-aware Multichannel Adaptive Framework for the IEEE 802.15. 4e-LLDN », *in*: *IEEE Transactions on Industrial Electronics* 63.*10* (2016), pp. 6360–6370.

[47] Achim Berger et al., « Improving IEEE 802.15. 4e LLDN Performance by Relaying and Extension of Combinatorial Testing », *in*: *2014 IEEE Emerging Technology and Factory Automation (ETFA)*, IEEE, pp. 1–4.

[48] Andreas Willig, Yakir Matusovsky, and Adriel Kind, « Retransmission Scheduling in 802.15. 4e LLDN-a Reinforcement Learning Approach with Relayers », *in*: *2016 26th International Telecommunication Networks and Applications Conference (ITNAC)*, IEEE, pp. 63–69.

[49] Masaki Takamori and Yasushi Yamao, « Enhancing Throughput of Star Topology Sensor Network by Group Acknowledgement Method and MCR SS-CSMA/CA », *in*: *2015 21st Asia-Pacific Conference on Communications (APCC)*, IEEE, pp. 343–347.

[50] Chen Chen et al., « GAS: A Group Acknowledgement Strategey in Internet of Vehicles », *in*: *2018 IEEE International Conference on Smart Internet of Things (SmartIoT)*, IEEE, pp. 1–8.

[51] Xavier Vilajosana et al., « IETF 6TiSCH: A Tutorial », *in*: *IEEE Communications Surveys & Tutorials* 22.*1* (2019), pp. 595–615.

[52] Thomas Watteyne, Ankur Mehta, and Kris Pister, « Reliability Through Frequency Diversity: Why Channel Hopping Makes Sense », *in*: *2009 6th ACM symposium on PE-WASUN*, pp. 116–123.

[53] Jonathan Muñoz et al., « Why Channel Hopping Makes Sense, Even With IEEE802. 15.4 OFDM at 2.4 GHz », *in*: *2018 Global Internet of Things Summit (GIoTS)*, IEEE, pp. 1–7.

[54] Yichao Jin et al., « A Centralized Scheduling Algorithm for IEEE 802.15. 4e TSCH Based Industrial Low Power Wireless Networks », *in*: *2016 IEEE WCNC*, IEEE, pp. 1–6.

[55] Maria Rita Palattella et al., « Traffic Aware Scheduling Algorithm for Reliable Low-power Multi-hop IEEE 802.15. 4e Networks », *in*: *2012 IEEE PIMRC*, pp. 327–332.

[56] Maria Rita Palattella et al., « On Optimal Scheduling in Duty-cycled Industrial IoT Applications Using IEEE802. 15.4 e TSCH », *in*: *IEEE Sensors Journal* 13.*10* (2013), pp. 3655–3666.

[57] Guillaume Gaillard et al., « High-reliability scheduling in deterministic wireless multi-hop networks », *in*: *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, IEEE, pp. 1–6.

[58] Guillaume Gaillard et al., « Kausa: KPI-aware scheduling algorithm for multi-flow in multi-hop IoT networks », *in*: *2016 International Conference on Ad-Hoc Networks and Wireless*, Springer, pp. 47–61.

[59] Vasileios Kotsiou, « Reliable Communications for the Industrial Internet of Things », PhD thesis, Université de Strasbourg, 2020.

[60] Tengfei Chang et al., *6TiSCH Minimal Scheduling Function (MSF)*, Internet-Draft draft-ietf-6tisch-msf-18, Work in Progress, IETF, Sept. 2020, 22 pp., URL: `https://datatracker.ietf.org/doc/html/draft-ietf-6tisch-msf-18`.

[61] Nicola Accettura et al., « Decentralized Traffic Aware Scheduling for Multi-hop Low Power Lossy Networks in the Internet of Things », *in*: *2013 IEEE 14th WoWMoM*, pp. 1–6.

[62] Tengfei Chang et al., « LLSF: Low latency scheduling function for 6TiSCH networks », *in*: *2016 International Conference on Distributed Computing in Sensor Systems (DCOSS)*, IEEE, pp. 93–95.

[63] Inès Hosni, Fabrice Théoleyre, and Noureddine Hamdi, « Localized scheduling for end-to-end delay constrained Low Power Lossy networks with 6TiSCH », *in*: *2016 IEEE Symposium on Computers and Communication (ISCC)*, IEEE, pp. 507–512.

[64] Ridha Soua, Pascale Minet, and Erwan Livolant, « WAVE: A Distributed Scheduling Algorithm for Convergecast in IEEE 802.15. 4e TSCH Networks », *in*: *Transactions on Emerging Telecommunications Technologies* 27.*4* (2016), pp. 557–575.

[65] Mohamed Osman and Frederic Nabki, « OSCAR: An optimized scheduling cell allocation algorithm for convergecast in IEEE 802.15. 4e TSCH networks », *in*: *Sensors* 21.*7* (2021), p. 2493.

[66] Ridha Soua, Pascale Minet, and Erwan Livolant, « DiSCA: A Distributed Scheduling for Convergecast in Multichannel Wireless Sensor Networks », *in*: *2015 IFIP/IEEE International Symposium on Integrated Network Management*, pp. 156–164.

[67] Simon Duquennoy et al., « Orchestra: Robust mesh networks through autonomously scheduled TSCH », *in*: *2015 13th ACM conference on embedded networked sensor systems*, pp. 337–350.

[68] Ren-Hung Hwang, Chih-Chiang Wang, and Wu-Bin Wang, « A distributed scheduling algorithm for ieee 802.15. 4e wireless sensor networks », *in*: *Computer Standards & Interfaces* 52 (2017), pp. 63–70.

[69] « IEEE Standard for Low-Rate Wireless Networks », *in*: *IEEE Std 802.15.4-2020 (Revision of IEEE Std 802.15.4-2015)* (2020), pp. 1–800, DOI: 10.1109/IEEESTD.2020.9144691.

[70] Tengfei Chang et al., « Adaptive Synchronization in Multi-hop TSCH Networks », *in*: *Computer Networks* 76 (2015).

[71] Georgios Z Papadopoulos et al., « Guard Time Optimisation and Adaptation for Energy Efficient Multi-hop TSCH Networks », *in*: *2016 IEEE 3rd WF-IoT*, IEEE, pp. 301–306.

[72] David Stanislowski et al., « Adaptive Synchronization in IEEE802. 15.4e Networks », *in*: *IEEE Transactions on Industrial Informatics* 10.*1* (2013), pp. 795–802.

[73] Tengfei Chang and Qin Wang, « Adaptive Compensation for Time-slotted Synchronization in Wireless Sensor Network », *in*: *International Journal of Distributed Sensor Networks* 10.*4* (2014), p. 540397.

[74] Alexandros Mavromatis et al., « Impact of guard time length on IEEE 802.15. 4e TSCH energy consumption », *in*: *2016 13th annual IEEE international conference on sensing, communication, and networking (SECON)*, IEEE, pp. 1–3.

[75] Georgios Z Papadopoulos et al., « Guard time optimisation for energy efficiency in IEEE 802.15. 4-2015 TSCH links », *in*: *Interoperability, Safety and Security in IoT*, Springer, 2016, pp. 56–63.

[76] Miao Xu et al., « Energy-efficient time synchronization in wireless sensor networks via temperature-aware compensation », *in*: *ACM Transactions on Sensor Networks (TOSN)* 12.*2* (2016), pp. 1–29.

[77] Borja Martinez, Xavier Vilajosana, and Diego Dujovne, « Accurate clock discipline for long-term synchronization intervals », *in*: *IEEE Sensors Journal* 17.*7* (2017), pp. 2249–2258.

[78] Vasileios Kotsiou et al., « Is local blacklisting relevant in slow channel hopping low-power wireless networks? », *in*: *2017 IEEE International Conference on Communications (ICC)*, IEEE, pp. 1–6.

[79] Dimitrios Zorbas, Georgios Z Papadopoulos, and Christos Douligeris, « Local or global radio channel blacklisting for ieee 802.15. 4-tsch networks? », *in*: *2018 IEEE International Conference on Communications (ICC)*, IEEE, pp. 1–6.

[80] Vasileios Kotsiou et al., « LABeL: Link-based adaptive blacklisting technique for 6TiSCH wireless industrial networks », *in*: *2017 20th ACM International Conference on Modelling, Analysis and Simulation of Wireless and Mobile Systems*, pp. 25–33.

[81] Alec Woo and David E Culler, *Evaluation of efficient link reliability estimators for low-power wireless networks*, Computer Science Division, University of California Oakland, CA, USA, 2003.

[82] Pedro Henrique Gomes, Thomas Watteyne, and Bhaskar Krishnamachari, « MABO-TSCH: multihop and blacklist-based optimized time synchronized channel hopping », *in*: *Transactions on Emerging Telecommunications Technologies* 29.7 (2018), e3223.

[83] Rasool Tavakoli et al., « Enhanced time-slotted channel hopping in WSNs using non-intrusive channel-quality estimation », *in*: *2015 IEEE 12th International Conference on Mobile Ad Hoc and Sensor Systems*, IEEE, pp. 217–225.

[84] Simon Haykin and Michael Moher, *Introduction to analog and digital communications*, Wiley, 2007.

[85] Bluetooth SIG Proprietary, « Bluetooth Specification Version 5.0 », *in*: (2016), pp. 1–2822.

[86] Leon W Couch, *Digital and analog communication systems*, vol. 7, Pearson.

[87] Dan Wolberg, Markus Rentschler, and Pascal Gaggero, « Simulative performance analysis of IO-link wireless », *in*: *2018 14th IEEE International Workshop on Factory Communication Systems (WFCS)*, IEEE, pp. 1–10.

[88] IO-Link Community, « IO-Link Wireless System Extensions v1.1 », *in*: (2018), pp. 1–302.

[89] Xinrong Huang et al., « Wireless smart battery management system for electric vehicles », *in*: *2020 IEEE Energy Conversion Congress and Exposition (ECCE)*, IEEE, pp. 5620–5625.

[90] Anif Jamaluddin et al., « Development of Wireless Battery Monitoring for electric vehicle », *in*: *2014 International Conference on Electrical Engineering and Computer Science (ICEECS)*, IEEE, pp. 147–151.

[91] Cody Shell et al., « Implementation of a wireless battery management system (WBMS) », *in*: *2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, IEEE, pp. 1954–1959.

[92] Akash Samanta and Sheldon S Williamson, « A Survey of Wireless Battery Management System: Topology, Emerging Trends, and Challenges », *in*: *Electronics* 10.*18* (2021), p. 2193.

[93] Ataur Rahman, Mizanur Rahman, and Mahbubur Rashid, « Wireless battery management system of electric transport », *in*: *IOP Conference Series: 2017 Materials Science and Engineering*, vol. 260, 1, p. 012029.

[94] Gereon Meyer, *Advanced Microsystems for Automotive Applications 2012: Smart Systems for Safe, Sustainable and Networked Vehicles*, Springer Science & Business Media, 2012.

[95] Yun Wu et al., « A Battery Management System for electric vehicle based on Zigbee and CAN », *in*: *2011 4th International Congress on Image and Signal Processing*, vol. 5, IEEE, pp. 2517–2521.

[96] Tadahide Kumtachi, Kazuhiko Kinoshita, and Takashi Watanabe, « Reliable wireless communications in battery management system of electric vehicles », *in*: *2017 Tenth International Conference on Mobile Computing and Ubiquitous Network (ICMU)*, IEEE, pp. 1–6.

[97] Minkyu Lee et al., « Wireless battery management system », *in*: *2013 World Electric Vehicle Symposium and Exhibition (EVS27)*, IEEE, pp. 1–5.

[98] Jaesik Lee et al., *Wireless battery area network for a smart battery management system*, US Patent 9,293,935, Mar. 2016.

[99] Marelli, *Marelli launches its Wireless Distributed Battery Management System*, 2022, URL: `https://www.marelli.com/marelli-launches-wireless-distributed-battery-management-system-wbms/` (visited on 06/30/2022).

[100] Renesas, *Wireless Battery Management Systems*, URL: `https://www.renesas.com/us/en/application/automotive/electrified-drivetrain-xev/wireless-battery-management-system` (visited on 06/30/2022).

[101] Mark NG Texas Instruments Incorporated, *Wireless Battery Management Systems*, 2021, URL: https://www.ti.com/lit/ml/slyp745/slyp745.pdf (visited on 06/30/2022).

[102] Texas Instruments Incorporated, *SimpleLink™ Microcontroller Platforms*, 2018, URL: https://www.ti.com/lit/ml/swat002b/swat002b.pdf (visited on 06/30/2022).

[103] David Lara and Danovan Porter, *Cable replacement using wireless technologies in automotive*, 2022, URL: https://www.ti.com/lit/ml/slyp670/slyp670.pdf (visited on 06/30/2022).

[104] Tomas Urban, « Functional Safety-Relevant Wireless Communication in Automotive Battery Management Systems », *in*: *Technical White Paper* (2022).

[105] Fabian Rincon, Yasuyuki Tanaka, and Thomas Watteyne, « On the impact of wifi on 2.4 ghz industrial IoT networks », *in*: *2018 IEEE International Conference on Industrial Internet (ICII)*, IEEE, pp. 33–39.

[106] Thomas Watteyne et al., « Technical overview of SmartMesh IP », *in*: *2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, IEEE, pp. 547–551.

[107] Greg Zimmer et al., « Wireless Battery Management Systems Highlight Industry's Drive for Higher Reliability », *in*: *Linear Technology, February* (2017).

[108] Analog Devices, *Wireless Battery Management Systems (wBMS)*, 2022, URL: https://www.analog.com/en/product-category/wireless-battery-management-systems.html (visited on 06/30/2022).

[109] *CC2652R SimpleLink™ Multiprotocol 2.4 GHz Wireless MCU*, CC2652, SWRS207H, Texas Instrument, Mar. 2021, URL: https://www.ti.com/lit/ds/symlink/cc2652r.pdf.

[110] Texas Instruments, *LAUNCHXL-CC26X2R1 SimpleLink™ multi-standard CC26x2R wireless MCU LaunchPad™ development kit*, 2022, URL: https://www.ti.com/tool/LAUNCHXL-CC26X2R1 (visited on 08/07/2022).

[111] Texas Instruments, *BQ79616-Q1, 16-S automotive precision battery monitor, balancer and integrated protector with ASIL-D compliance*, 2022, URL: https://www.ti.com/product/BQ79616-Q1 (visited on 08/07/2022).

[112] Joakim Lindh et al., *Measuring CC13xx and CC26xx current consumption*, Application Report, Texas Instrument, Jan. 2019, 38 pp., URL: https://www.ti.com/lit/an/swra478d/swra478d.pdf.

[113] Fatima Zahra Djiroun and Djamel Djenouri, « MAC Protocols with Wake-up Radio for Wireless Sensor Networks: A review », *in*: *IEEE Communications surveys & tutorials* 19.*1* (2016), pp. 587–618.

[114] Sebastian L Sampayo et al., « Is Wake-up Radio the Ultimate Solution to the Latency-energy Tradeoff in Multi-hop Wireless Sensor Networks? », *in*: *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications*, IEEE, pp. 1–8.

DOCTORAT BRETAGNE LOIRE / MATHSTIC

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Titre : Communication sans fil robuste pour les systèmes de gestion de batterie dans les véhicules électriques

Mots clés : Véhicule Électrique, Système de gestion de la batterie sans fils, GACK, TSCH, BLE

Résumé : Le système de gestion de la batterie (BMS) est chargé de surveiller et garantir le bon fonctionnement de la batterie dans un Véhicule Électrique (VE). Le BMS utilise une architecture maître-esclave avec une liaison filaire entre les capteurs et l'unité de calcul. Dernièrement, l'industrie a vu l'intérêt de remplacer ce bus filaire par une liaison sans fil. Le BMS sans fil (WBMS) peut apporter de nombreux avantages, mais aussi plusieurs défis. Dans cette thèse, nous présentons une solution complète qui rend possible la communication sans fil robuste dans le BMS. Nous proposons une amélioration du protocole d'accès au milieu « Time Slotted Channel Hopping » (TSCH) en conjonction avec la couche physique de Bluetooth Low Energy (BLE). Ce réseau est géré par une fonction d'ordonnancement fiable et prévisible basée sur la méthode d'accusé de réception de groupe (GACK). Nous proposons aussi un mode endormi pour le WBMS qui reconfigure l'architecture du réseau permettant au nœud central de s'éteindre à fin d'éviter la décharge de la petite batterie 12V du VE. Nous discutons également des nouvelles opportunités que l'inclusion d'une capacité de calcul dans les esclaves apporte au développement logiciel du BMS. Les résultats des tests en laboratoire et sur véhicule montrent que notre système offre une haute fiabilité, une latence spécifique et une faible consommation d'énergie.

Title: Enabling Robust Wireless Communication for Battery Management Systems in Electric Vehicles

Keywords: Electric Vehicle, Wireless Battery Management System, GACK, TSCH, BLE

Abstract: The Battery Management System (BMS) is in charge of monitoring and ensuring the correct operation of the Electric Vehicle (EV) battery. The BMS uses a master slave architecture with a wired daisy chain link. Lately, the industry has seen interest in replacing the wired bus with a wireless network. Wireless BMS (WBMS) brings multiple advantages but at the same time several challenges. In this thesis, we present a complete solution to enable robust wireless communications inside the battery pack of EVs. We propose an enhanced version of the IEEE Std 802.15.4 Time Slotted Channel Hopping (TSCH) Medium Access Control (MAC) mode running over the physical layer of Bluetooth Low Energy (BLE). The network is managed by a reliable and predictable scheduling function based on the Group Acknowledgement (GACK) method. For the parking state of the vehicle, we propose a WBMS sleep mode that reconfigures the network architecture allowing the central node to switch off, preventing the discharge of the small 12 V battery of the EV. We also discuss the new opportunities that the inclusion of a computing capacity in the slaves brings to BMS software development. The laboratory and vehicle test results show that our system provides high reliability, bounded latency, and low power consumption.