Università degli Studi di Padova

Department of Information Engineering

Master Thesis in Master in ICT for Internet and Multimedia

Experimental Study on Real-Time Wireless Networks for Motion Control of Manipulator and Mobile Platform In Industrial Robotics

Supervisor
Stefano Vitturi
Università di Padova

Co-supervisor
Michele Luvisotto
Zhibo Pang
ABB Corporate Research

Master Candidate
Marco Rossanese

Examiner
Federico Tramarin

Padua, 2nd of December 2019
Academic Year 2018 - 2019

Marco Rossanese

Prof. Stefano Vitturi

”Success never goes on sale. The cost is always blood, sweat and tears.”

Eric Thomas

Abstract

Intelligent manufacturing is currently an hot topic, it promises to increase profits, productivity and safety. The integration of ICT with manufacturing technologies is one of the key steps towards this target. Many industrial wireless control networks (IWCN) have been developed to provide efficient networking, but no protocol has established itself as principal solution. Some of the main challenges that IWCNs aim to solve are supporting mobile equipment as robots in critical motion control and providing a complete integration with the present wired communications. Regarding these topics, the literature presents several theoretical approaches but very few papers reporting practical applications. The goal of this thesis is to investigate some industrial application scenarios and evaluate the performance of selected wireless technologies in these contexts. Most of the available solutions are not ready to satisfy the necessary requirements as high reliability, real-time and low jitter, but some of them are promising. A recently standardized industrial wireless technology, WIA-FA, has shown good performance in practical deployments of selected use cases. Two experimental applications are considered: first, the EGM path planning library is tested with different wireless technologies, then a CANbus communication is bridged with WIA-FA for the control of mobile platforms. The experimental evaluations show that WIA-FA meets the requirements of the considered scenarios.

Contents

Abstract vii

List of figures xi

List of tables xiii

1 Introduction 1

2 Applications and requirements 5
- 2.1 Use cases . . . 6
- 2.2 Mobile robot platforms . . . 8
- 2.3 Communication requirements . . . 11
- 2.4 Communication interfaces . . . 13
  - 2.4.1 EGM for manipulators . . . 13
  - 2.4.2 CAN for mobile platforms . . . 16

3 State of the art 21
- 3.1 Wireless for industrial mobile robotics . . . 22
  - 3.1.1 Architecture . . . 22
  - 3.1.2 Available wireless solutions . . . 24
- 3.2 Related work . . . 33
  - 3.2.1 Wireless for motion control . . . 33
  - 3.2.2 Wireless-CAN integration . . . 34

4 WIA-FA overview 35
- 4.1 The WIA-FA system . . . 36
  - 4.1.1 Devices . . . 36
  - 4.1.2 Topology and Management . . . 36
  - 4.1.3 Protocol stack . . . 37
- 4.2 Proprietary WIA-FA technologies . . . 41
  - 4.2.1 Superframe allocation . . . 41
  - 4.2.2 Retransmission . . . 42
  - 4.2.3 Frame aggregation . . . 43
- 4.3 Practical applications . . . 44

5 Wireless EGM 47
5.1 Architecture . . . 48
5.2 Setups . . . 50
5.2.1 WIA-FA workaround . . . 54
5.3 Results . . . 55
5.3.1 Performance evaluation with a simulated robot . . . 56
5.3.2 Performance evaluation with a real robot . . . 58
5.3.3 WIA-FA layer 2 mode communication . . . 61
5.4 Conclusion . . . 63

6 Wireless CAN 65
6.1 Setup components . . . 66
6.1.1 Controllers . . . 66
6.1.2 Xilinx System on Chip . . . 67
6.2 Setup architecture . . . 70
6.3 Results . . . 73
6.3.1 Point to point architecture . . . 73
6.3.2 Point to multipoint architecture . . . 78
6.3.3 Experiment by emulated traffic . . . 85
6.4 Conclusion . . . 88

7 Future technologies 89
7.1 5G . . . 90
7.1.1 Requirements and enabling technologies . . . 91
7.1.2 5G Architecture . . . 94
7.1.3 3GPP standardized slices . . . 96
7.1.4 URLLC technology components . . . 98
7.2 IEEE802 . . . 99
7.2.1 IEEE802.11ax . . . 99
7.2.2 IEEE802.11be . . . 105
7.3 Wireless TSN . . . 107
7.3.1 Time sensitive networking . . . 108
7.3.2 Challenges and gaps for wireless TSN . . . 111

8 Conclusion 115
8.1 Future work . . . 117

Appendix A Circuit schematics and wiring 119

References 123

Acknowledgments 129

Listing of figures

2.1 Mobile robots and manipulators . . . 7
2.2 AGV examples . . . 8
2.3 Mobile platforms . . . 9
2.4 EGM UdpUc . . . 14
2.5 CAN protocol stack . . . 16
2.6 Format of CAN data frames . . . 18

3.1 Industrial Scenario . . . 22

4.1 Topology . . . 37
4.2 WIA-FA stack . . . 38
4.3 WIA-FA superframe . . . 39
4.4 Beacon allocation example . . . 41
4.5 GACK retransmission mode . . . 43
4.6 AVGs Synchronization . . . 45
4.7 AVGs sorting system architecture . . . 45

5.1 Proposed architecture for EGM . . . 48
5.2 Ethernet based EGM implementation . . . 50
5.3 Commercial WiFi based EGM implementation . . . 51
5.4 LTE based EGM implementation . . . 51
5.5 WIA-FA based EGM implementation . . . 52
5.6 WIA-FA and LTE lab setups . . . 53
5.7 Ethernet vs. WIA-FA, RobotStudio . . . 56
5.8 Commercial wifi . . . 57
5.9 Ethernet vs. WIA-FA vs. LTE, YuMi . . . 58
5.10 Cumulative Distribution Function for WIA-FA timeslots . . . 59
5.11 CDF RTT for general comparison . . . 60
5.12 Layer 2 mode simulation setup . . . 61
5.13 RTT layer 2 . . . 62

6.1 F3 and F7 controllers . . . 66
6.2 Xilinx Zynq-7000 SoC ZC706 . . . 67
6.3 Point to point architecture . . . 70
6.4 Point to multipoint architecture . . . 71
6.5 Point to multipoint lab setup . . . 72

6.6 Half-duplex communication . . . 73
6.7 Pseudocode for duplex transmission . . . 75
6.8 Dumped packets with one F3 slave, full duplex communication . . . 76
6.9 Arrival times for point to point architecture . . . 77
6.10 Dumped packets with one F3 slave, full duplex communication, 4 packets merged at master . . . 78
6.11 Packet arrival times for point to multipoint architecture . . . 79
6.12 Dumped packets with 4 F3 slaves, critical IDs only . . . 80
6.13 Dumped packets with 4 F3 slaves, four packets merged at slave . . . 82
6.14 Dumped packets with 4 F3 slaves, five packets merged at slave . . . 83
6.15 Adaptive solution scheme . . . 84
6.16 Dumped packets with 4 F3 slaves, adaptive solution . . . 84
6.17 Representation of the emulated traffic flows . . . 86
6.18 Dumped packets with 4 F3 slaves, emulated traffic . . . 87

7.1 Network slicing in 5G networks . . . 91
7.2 5G System architecture . . . 94
7.3 5G and 4G comparison and slices comparison . . . 97
7.4 OFDMA example . . . 101
7.5 Example of OFDMA transmission in IEEE802.11ax . . . 102
7.6 An example of UL OFDMA transmission . . . 103
7.7 Classification of TSN standardization . . . 108
7.8 gPTP domain in IEEE802.11as . . . 109
7.9 TSN protocol stack . . . 111
7.10 Integrating TSN in 5G architecture . . . 112

A.1 F3 circuit schematic . . . 119
A.2 Transceiver and IC adapter and DSUB9 port . . . 120
A.3 Female D-sub 9 wiring . . . 121
A.4 CAN pin outputs . . . 121
A.5 Male D-sub 9 wiring for F7 bus . . . 122
A.6 Male D-sub 9 wiring for F3 bus . . . 122

Listing of tables

3.1 Advantages of possible architectures . . . 23
3.2 Relevant IEEE802.11 amendment . . . 24
3.3 Wireless technologies comparison . . . 31

4.1 WIA-FA physical layer parameters . . . 38
4.2 Payload format of an aggregated frame . . . 44

5.1 Statistics for RobotStudio comparison . . . 56
5.2 Statistics for YuMi comparison . . . 58
5.3 Statistics of the RTT for different timeslot . . . 59
5.4 Statistics of the RTT for EGM communication through WIA-FA, LTE and Ethernet . . . 60
5.5 Statistics of the RTT for layer 2 mode . . . 62

6.1 Packets generated by F3 . . . 74
6.2 Packets generated by F7 . . . 74
6.3 Performance with one F3 slave, bidirectional communication . . . 76
6.4 Performance with one F3 slave, bidirectional communication, four merged packets at master . . . 78
6.5 Packets generated by the second F3 . . . 79
6.6 Packets generated by the third F3 . . . 79
6.7 Packets generated by the fourth F3 . . . 79
6.8 Performance with 4 F3 slaves, critical IDs only . . . 81
6.9 Loss rate with 4 F3 slaves, critical IDs only, for different WIA-FA timeslot lengths . . . 81
6.10 Performance with 4 F3 slaves, 4 packets merged at slave . . . 82
6.11 Performance with 4 F3 slaves, 5 packets merged at slave . . . 83
6.12 Performance with 4 F3 slaves, adaptive solution . . . 85
6.13 Loss rate with 4 F3 slaves, adaptive solution, for different WIA-FA timeslot lengths . . . 85
6.14 Performance with 4 F3 slaves, emulated traffic . . . 86
6.15 Loss rate with 4 F3 slaves, emulated traffic, for different WIA-FA timeslot lengths . . . 87

7.1 Low latency and high reliability for different use cases . . . 90

Do what you can, with what you have, where you are.

Theodore Roosevelt

1 Introduction

Nowadays intelligent manufacturing is a trending topic, attracting a huge interest from researchers, governments and manufacturers. It promises to reduce the energy consumption, increase economic benefits, enable customized production and last but not least increase the safety in human-machine working environment.

The integration of information and communication technologies (ICTs) with manufacturing technologies is one of the key steps towards this vision. Industrial wireless control networks (IWCNs) have been developed to provide an efficient networking support to mobile equipment as robots, track-mounted equipment, rotary equipment and mobile assets. Besides making monitoring and control easier, ICTs aim to achieve the more ambitious goal of fully replacing wired connections in industrial manufacturing environments.

To satisfy the strict industrial communication requirements like high reliability, hard real-time and low jitter, several efforts have been devoted to the development of IWCNs standards. However there is still a lot of work to do, especially from the ICT side, in order to establish intelligent manufacturing as a solid reality.

Currently no wireless protocol has established itself as main solution in the industrial field, because the lack of reliability of wireless network is seen as a dangerous risk for such applica-

tions. The committees and the organizations controlling the principal wireless technologies view the gap present in this sector as a big opportunity. Indeed, in the next few years, many new technologies are expected to provide solid solutions to bridge this gap by offering higher performance.

Some of the main challenges that the research in this field is facing are the realization of a wireless network able to support critical motion control for robotics and, as already anticipated, the complete integration of the present wired communication technologies with the wireless ones, guaranteeing determinism and real time. Regarding these topics, the literature presents several theoretical approaches but very few papers reporting practical applications.

The goal of this thesis is to investigate some industrial application scenarios and evaluate the performance of selected wireless technologies in these contexts. Most of the available solutions are not ready to satisfy the necessary requirements as high reliability, real-time and low jitter, but some of them are promising. A recently standardized industrial wireless technology, WIA-FA, has shown good performance in practical deployments of selected use cases.

In this thesis, two practical applications are considered:

- First, the Externally Guided Motion library, used for external planning of movements and paths for an ABB robot, is tested with different communication technologies. These assessments are made by means of a simulated environment as well as on an ABB robot, YuMi.
- Secondly, a preliminary experimental research study is presented in which a CANbus communication is bridged with WIA-FA for applications concerned with the control of mobile platforms. Different possible bridging strategies are considered and the relative performance evaluations are given.

The methodology employed in the experimental sections can be summarized in this process flow: a deep information research is the first stage, followed then by the development of multiple options. At this stage, the best option based on technical requirements is chosen and an hardware/software design is outlined. The solution is finally tested, highlighting strengths and weaknesses.

From the experiments, it results that WIA-FA is a promising standard for industrial applica-

tions, since it provides a reliable medium access technology for guaranteeing real-time communications and the ability to support a refresh rate around 2 ms. Such performance can be accounted good enough for the integration with some of already established industrial protocols. Moreover it showed position errors comparable with other wired technologies, when dealing with motion control applications.

It is necessary to remark that all the assessments performed are limited to a point to point scenario in a laboratory environment. It will be part of the future work to extend the evaluation to multiple nodes and in a more realistic industrial scenario.

This thesis is structured as follows: in Section 2, an overview of industrial use cases is given with a further description of those involved in the experimental studies. In Section 3, the state of art of the main industrial wireless technologies is discussed together with the related works in the literature, followed, then, by the WIA-FA standard in Section 4. Subsequently, the thesis describes the two experimental evaluations: in Section 5 the performance of EGM is benchmarked for different communication technologies and, in Section 6, a novel integration of WIA-FA standard over CANbus is shown. In Section 7, an overview of future technologies that can be promising for the described use cases. Finally, in Section 8, conclusions are presented together with future work.

Two roads diverged in a wood, and I — I took the one less traveled by, and that has made all the difference.

Robert Frost, The Road Not Taken

2 Applications and requirements

Integrating ICTs with manufacturing technologies is one of the main challenges in smart manufacturing. In such a scenario, industrial robots will ease many different operations and industrial networks will allow to control them remotely.

The primary goal in these applications is to establish a safe environment in which human workers and machines can coexist without endangering each other. Therefore the control of robots is fundamental to maintain a correct behavior of the machines and it must be realized by means of high-performance industrial networks. Although wired industrial networks, such as fieldbuses and industrial Ethernet, are mostly employed to control robots today, the use of IWCNs is strongly advocated to reduce the costs and increase the flexibility, unlocking new applications, such as mobile robotics.

For these reasons, a strong research effort is currently put in the design and implementation of high-performance IWCNs. Currently available solutions are still far from reaching the desired performance and are hence still seen as an high risk investment. However recently released and currently planned ICT technologies are rapidly bridging the gap, targeting the ambitious goal of fully replacing wired connections in industrial manufacturing environments of the future.

2.1 Use cases

There are several use cases for wireless communication in industrial manufacturing and logistics, but also in service-related fields, for example hospital logistics. These cases can stretch from the control of one mobile robot to several robots and from periodic monitoring of machine status to high-speed control of mobile devices.
The following are some typical use cases that explicitly require wireless communication:

- Control of a mobile robot;
- Control of automated guided vehicles (AGVs) fleet;
  - Small fleet (<10 AGVs);
  - Large fleet (10-100 AGVs);
- Synchronized control of collaborating AGVs;
- Machine collaboration;

Starting from the top, a mobile robot is constituted of a moving platform and a manipulating arm placed on top of it, as shown, for example in Fig. 2.1b. A simultaneous control of the basis and the arm must be guaranteed. Such a platform is usually employed to grasp a wide range of objects with different sizes within a certain weight and move them to a specified location. The main communication challenges are reliability and latency, resulting in the inability to handle dynamic environments without compromising speed or human safety. Moreover a complete integration between wireless network and onboard wired network must be ensured. These topics will be deeply discussed in chapters 5 and 6.
Controlling a small fleet of AGVs implies the accomplishment of numerous and different tasks by each of them independently: AGVs have the ability to move at different speeds (0.5-3 m/s) following their own route without interfering with each other's operation. The main challenge in this use case is to ensure reliable and uninterrupted real-time communication between the AGVs and the central controller. [2]

Increasing the number of AGVs to a few hundreds is even more challenging, because dangerous situations are more likely to occur. Furthermore, scalability is a fundamental issue for large fleets: scalable solutions are desired for cost reduction and quick resizing of the fleet.

(a) (b)

Figure 2.1: (a) Mobile robot and manipulator, (b) ABB mYuMi taken from [1].

Another fascinating use case is the AGV synchronization, in which a little group of small AGVs, performs a cooperative task as carrying a large and heavy component. Cooperation of multiple robots will play an important role in intelligent manufacturing, to complete heavy tasks quickly enhancing productiveness and safety. However, unreliable and not secure wireless communication is often the reason for not deploying AGVs in the factories/warehouses and the lack of well-tested cooperative algorithms is another reason why this is hard to put into practice.

Machine collaboration will be another key feature in intelligent manufacturing because it aims to make several robots or machines with different functionality (e.g conveyor belt, sensors, stationary robots, controllers, AGVs, mobile robot, etc.) working together in order to reach the predefined goal. In this use case, the task to be accomplished can be split in subtasks to assign to many machines to be executed in the most effective way. The lack of machine collaboration and the fragmentation of production lines in disjointed parts is nowadays a pain point for many industries, which could benefit from this use case. The communication requirements for this scenario are very high, since if a section of the process line stops or slows down, the throughput of the whole line is consequently affected.

2.2 Mobile robot platforms

It is well known that robotics encompasses a wide range of devices with different features and purposes, but in this thesis only the mobile ones will be taken in account.

Up to now, there is not any standard definition for "mobile robot", but as the name suggests, these are machines that can move from one place to another autonomously [3]. Robots aim to accomplish their tasks without any assistance of human operators. The main advantage is that they can move freely in a predefined workspace and this makes them useful for several applications in both structured and unstructured environments.
For example, mobile robots used in factories are automated guided vehicles. In military operations they are categorized as unmanned ground reconnaissance vehicles, then they can be exploited in healthcare to deliver pharmaceutical items, in search and rescue and also in our daily life for lawn mowing and floor cleaning.

AGVs are the principal category of mobile robots treated in this thesis. They were introduced in 1953, but a rapid development began only in the 1980s thanks to the stunning progresses in computer sciences, enhancing the reliability by means of efficient embedded systems, in mechanics, producing more effective motors, and in electronics improving the power consumption. Fig. 2.2 shows typical AGVs, like tuggers usefull to pull carts, unit loaders for parts-tray transfers and fork trucks to lift heavy packages or objects.

Figure 2.2: AGV examples: from the left, a unit loader, a tugger and a fork lift.

For what concerns the mobile platforms, an excellent overview of the state of art and a valuable market analysis are presented in [4].

In this investigation, focused on for new low cost driveline designs, it was concluded that industrial platforms are conceived only for smooth surfaces, therefore, whenever the robots need to face an uneven surface with cracks, bumps or similar features, a motion failure may occur. The only platforms able to cope with these irregularities are extreme off-road military robots which are rather expensive.

The best solutions that can been found in the market all exploit omni-directional wheel drives, that is drives which do not need a steering mechanism because the orientation of rollers on the wheels allows to slide and slip according to the wanted direction; an example is shown in Fig. 2.3a. Few popular mobile platforms are now listed below:

(a) Omni-directional Drive.

(b) RidgeBack.

(c) KMR Quantec.

(d) MiR 100.

Figure 2.3: Mobile platforms present on the market.

RidgeBack, Fig. 2.3b, is a compact and very customizable platform; KMR Quantec by KUKA,

Fig. 2.3c, is a robust heavy-duty platform equipped with LiDAR sensors to retrieve its position accurately; MiR 100 by Mobile Industrial Robotics, Fig. 2.3d, is a small platform equipped with six tyres to make it more agile.

It must be clear now that there is no common approach for every kind of application; each sort of scenario and task strongly affects the design, the features to realize and how they are put into practice by the robot.

It is necessary to introduce the concept of robot control which deals, for example, with the problem of determining the torques and forces that must be developed by the robotic actuators to reach a desired position, track a desired trajectory and in general to perform tasks with specific performance requirements. It is hence essential to properly design the robot control system, depending on the goal to carry out, on the environment, regarding its space dimension and if it is structured or unstructured, and the on-board equipment, which determines the computing power and the ability to sense the surroundings.

2.3 Communication requirements

The main technical requirements for a wireless network used for control of AGVs or mobile robots can be summarized as follows:

1. Periodical exchange of control data with low cycle times (down to a few milliseconds).
2. Deterministic data delivery.
3. High reliability in data exchange.
4. Support for mobility at low speeds (up to a few m/s).
5. Support for a large number of nodes (up to a few hundreds).
6. Coverage of a wide area (up to 200x200 m).
7. Seamless integration with industrial wired networks.
8. Secure data exchange.

Some of these requirements only apply to selected use cases, but, in general, a solution which supports all the requirements simultaneously is preferable.

Guaranteeing periodical exchange of control data every few milliseconds is crucial for maintaining safety in the workspace. Even though there are no performance standards for mobile robots, national and international safety standards have been outlined by various committees and must be respected. A list of most relevant mobile robot safety standards for US and Europe is shown in [5].

Determinism is a fundamental feature in IWCNs and it is needed because industrial applications require updated information to be available exactly at certain times in order for the actuation commands to be computed and distributed. Therefore, in industrial communications, periodic messages must be delivered before a time deadline and the fluctuations in the periodicity (jitter) must be as low as possible. Generally speaking, determinism can be achieved adopting a proper medium access mechanism.

High reliability is a requirement that must be fulfilled in industrial applications because safety-critical messages are often involved. Alarms are the clearest example, a loss can cause

serious hazards to people and equipment. A good medium access scheme implies that collisions are prevented during packets transmissions, however this is not enough, because interference path loss and fading can lead to signal degradation and unsuccessful transmissions.

Since AGVs can move from an access point to another and the route they follow can be changed on-the-fly, there is no way to predict their location in advance. This creates problem to the wireless network, especially in terms of handover times between different access points that can affect the required latency.

The typical environments are warehouses or production facilities and usually they are quite big spaces, requiring a large coverage area. Moreover these facilities are harsh environments with several static/non-static metallic obstacles creating blind spots which degrade the signals.

Most of the communication infrastructures in industrial scenarios are wired, so migrating to the wireless solutions cannot be drastic, it has to be made as a step by step process. Moreover compatibility with the previous and already installed wired protocols such as Ethernet, Powerlink, Profinet, EtherCAT, etc must be guaranteed.

In wired communications is impossible to eavesdrop wired communications without having access to the end terminals. When using wireless, the nature of the medium is shared and special measures must be taken to avoid intruders and the most common cyberattacks. Moreover, the encryption should not be too complex to add only a little overhead on the data packets.

The most common solution for the wireless control of AGVs nowadays is arguably the WiFi technology, based on the IEEE802.11 standard. However, while WiFi can easily handle the asynchronous and low-rate data exchange of distributed control architectures, it has still several issues to be solved. Particularly, it does not provide a deterministic access scheme, it suffers interference/non-line-of-sight and integration with real-time wired solution is typically not feasible. This topic is discussed more in detail in section 3.1.2.

2.4 Communication interfaces

The communication requirements outlined in section 2.3 are difficult to obtain when using a wireless network. Two different communication interfaces, used to control robot movements and mobile platforms, are individually investigated when realized over a wireless network. In detail, the first one is Externally Guided Motion (EGM), a proprietary protocol created by ABB to control manipulators from an external device via UDP streaming. The second interface is Control Area Network (CAN), a wired fieldbus commonly used in automotive industries to control distributed drive units and recently adopted in in-vehicles control for mobile robots. These two interfaces are described in detail in the remaining of this chapter.

2.4.1 EGM for manipulators

Externally Guided Motion is an interface created by ABB to control robots in a low-level manner and it is defined in [6]. The Robot Controller (RC) typically performs both path planning and low-level joint control. EGM can completely by-pass the path planning in order to provide information about the robot at a high rate (250 Hz). In order to do that, external sensors are connected to the RC through Ethernet, creating what is called sensor fusion. In this way, a wider knowledge of the environment is obtained, which can lead to a revised or completely new path for the robot.

Externally Guided Motion provides two different features, Position Guidance and Path Correction.
In the former, the robots do not follow the programmed path, but one generated by an external device and the robots will be moved to the specified position. Since the path planning is by-passed by EGM in the robot controller, the robots will react quickly to all position references sent to the controller, even the faulty ones. The time between writing a new position until that given position starts to affect the actual robot position is around 20 ms.
Some examples are placing an object at a location that was given by an external sensor or picking objects from a bin using an external sensor to identify the object and retrieve the position.

In the latter, the programmed path for the robots is corrected using measurements furnished by an external device, so robots will be moved along the revised path. An example is tracking of objects in the surroundings of the known path.

Major limitations for EGM are that it can only be used on 6-axis robots and only with RAPID tasks, an high level programming language used for ABB robots; moreover only one external device can be used per robot to provide correction data.

UdpUc and sensor protocol

In general a scenario, EGM interface and motion control are integrated in the robot controller, so this communication cannot be investigated. The sensor is an external device, which is typically connected with EGM through Ethernet. During the application of this interface, no real sensor was used because its functions were emulated by an industrial PC.

The communication between the robot and the sensor is put in place by means of a protocol called User Datagram Protocol Unicast Communication (UdpUc). The data flow is illustrated in Fig. 2.4.

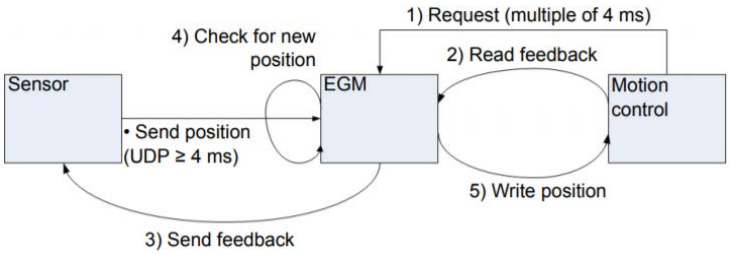


Figure 2.4: User Datagram Protocol Unicast Communication data flow. [6]

1. Motion control calls EGM.
2. Feedback data from motion control is red by EGM.
3. EGM sends feedback data to the sensor.
4. EGM checks queue if new UDP messages from the sensor are present.
5. If affirmative, EGM reads the message and writes the position data to motion control. Otherwise, motion control uses the latest position data previously written by EGM.

UdpUc is based on UDP in order to provide as real-time performance as possible; furthermore it can be implemented on several communication technologies, for instance, Ethernet which is the most used, WiFi, LTE, etc. The EGM is using Google Protocol Buffers for packet

encoding because allows to serialize/deserialize data in a very efficient way and is language-neutral.

The sensor is acting as a server and it cannot send anything to the robot before it has received a first message from the robot controller. Messages can be sent independently of each other in both directions after that first message.

The EGM sensor protocol data structures are defined by the EGM proto file. The proto file is compiled generating a serialized code, the application reads a message from the network, runs the deserialization, exploits the data and so the way back.

2.4.2 CAN for mobile platforms

The second use case aims to bridge a CANbus with a wireless network. Control Area Network, also know as CANbus or simply CAN, is one of the leading fieldbuses, it was developed in the 80s and adopted mainly as in-vehicle network for electronics and engine management. Only the physical and the data link layers have been standardized, multiple application layers with different purposes were presented, but none of them really established itself. Since for the involved use case real-time is a fundamental requirement, only the two bottom layers are adopted and they are discussed in the following.

Protocol stack

The CAN protocol stack is structured according to the ISO/OSI model, it includes only few layers to make implementations more efficient and inexpensive. The architecture is exposed in Fig. 2.5.

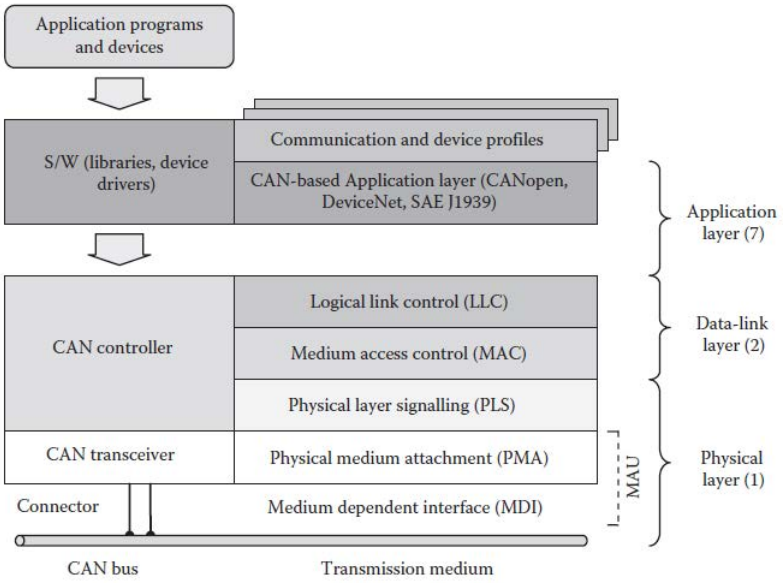


Figure 2.5: CAN protocol stack. [7]

The physical layer manages the transmission of information over the medium and is related to the electrical aspects. It is divided in physical layer signaling (PLS), physical medium attachment (PMA) and medium dependent interface (MDI). PLS takes care of bit representation, timing, and synchronization which are embedded into the CAN controllers. PMA deals with transmission/reception on the bus, detects bus failures and is connected to the medium

(the CAN bus) through MDI, a mechanical and electrical interface. Both are mostly implemented in the transceivers.

The data link layer, exploits the PLS services to transfer frames encoded as bit sequences. It is split into medium access control (MAC) and logical link control (LLC). MAC functions include frame encapsulation and decapsulation, arbitration, signaling and error checking. The LLC offers the user an interface for the communication services and decides if an incoming message is relevant for the node, in fact, frames do not have a destination address.

Physical layer

CAN is based on a bus topology which must be terminated with 120 Ω resistors at each end to suppress signal reflections.

Several kinds of transmission media can be used, single/double-wire buses for cheap implementations as well as optical fibers to ensure immunity to electromagnetic noise.

Several bit rates can be selected in the range from 50 kbit/s to 1 Mbit/s, since CAN does not mandate a specific value. The maximum extension of a CAN depends directly on the bit rate: the product between the bit rate and bus length has to be approximately constant. The reference is that the maximum extension allowed for a 500 kbit/s network is about 100 m.

Medium access technique

The CAN MAC mechanism is the CSMA scheme. Before transmitting a node has to sense the network, if it is idle, the frame transmission begins immediately, otherwise, the node must wait for the end of the current frame transmission.

One node might start its transmission while another frame is already travelling on the bus: due to the propagation delay in the medium, the node can sense the network as free even if a transmission has already begun, generating a collision. The probability for a collision to occur depends on to the number of nodes and their message generation rate.

CAN is able to resolve a contention in a deterministic way thanks to the arbitration scheme, so that time and bandwidth are not wasted. Indeed, a station stops the transmission of a

frame when it realizes that a frame of higher priority is being transmitted by another station. Frame priority is determined by the CAN Identifier (ID).

Since the ID is a sequence of bits, the "binary countdown" technique is employed to resolve the collision. Starting from the most important bit, the ID of involved frames are compared and those that present a bit equal to 1, loses the contention. Therefore, all the losers will retry the transmission when exchange of the current frame ends.

It is clear that each ID must be unique in the network, otherwise the arbitration scheme would lead to unmanageable situations.

Frames

The CAN protocol exploits four kinds of frames, namely, data, remote, error and overload frames. The ID for these frames can have two formats, the base which adopts 11-bit, so up 2048 different IDs are available, and the extended which assigns 29 bits, up to half a billion different objects could exist.

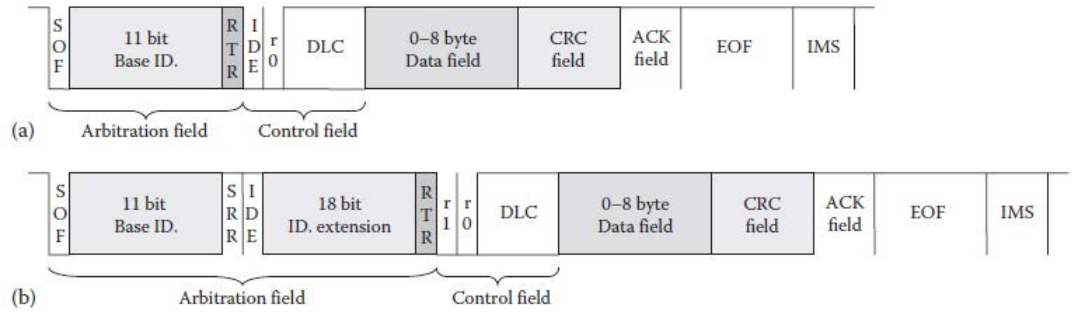


Figure 2.6: Format of CAN data frames: (a) base format and (b) extended format. [7]

Data frames are used to send information over the network. Each CAN data frame begins with an SOF bit to mark the beginning of the frame to synchronize the receiving nodes. It is followed by the arbitration field, which is different for both base and extended format. The former includes ID field and the remote transmission request (RTR) bit, used to discriminate between data and remote frames. The latter, includes also the substitute remote request (SRR), just a placeholder bit to preserve the structure, and the identifier extension (IDE) bit to discriminate between the formats.

The control field comes next to the arbitration and it comprises 6 bits. For base frames it consists of IDE, a reserved bit r0 and 4 bits for the data length code (DLC), which specifies the length (in bytes) of the data field. For the extended format, instead of IDE a reserved bit r1 is placed to maintain the structure.

The 8 bits data field stores the payload of the frame, then the cyclic redundancy check sequence (CRC) and acknowledgment (ACK) fields follow up. The CRC is encoded on 15 bits, the 2 bits ACK are used to discover if at least node of the network received the frame correctly.

The end of frame (EOF) field of 7 bits closes the frame notifying all the nodes the end of an error-free transmission. Frames are interleaved by the interframe space, the intermission (IMS), in which no transmission takes place.

In real applications, the total length for both formats is 16 bytes, indeed for the base a padding is necessary. An overview of a CAN data frame for both the formats is shown in Fig. 2.6.

Remote frames data are used to request the transmission of a given message by a remote node and they do not carry data. The requesting node does not know who is the producer of the relevant information and it is up to receivers to discover who has to reply, checking the ID field in the remote frame.

Error frames notify nodes about error occurrences and overload frames can be used by slower receivers to slow down operations in the network.

Labor omnia vincit improbus.

Virgil, Georgics, I, 145

3 State of the art

In this section the main wireless standards that can be exploited for industrial purposes are presented, highlighting their strengths and weaknesses, then the most relevant works regarding the use cases involved in this thesis are discussed.

Up to now, several international standards have been published by different organizations in order to guarantee interoperability among different vendors for the same technology and to share the frequency spectrum in the fairest way possible. Since the choice of using a specific technology strongly affects performance, costs and interoperability, it must be pondered beforehand in relation with the goal to achieve in each specific scenario.

Wireless standards can cover licensed or unlicensed portions of the spectrum: the licensed ones permit exclusive operations and, as consequence, guarantee no interference from external sources, at the cost of a fee to telecommunication operators. The most common choice for industrial solution, however, is the unlicensed spectrum, free under specific rules, but in densely populated area it is very likely that these bands of spectrum are crowded so much to cause interference.

It is also noteworthy the recent trend of deploying private cellular networks in which a telecom operator subleases a slice of their licensed spectrum to an enterprise to install its own network, allowing easier compliance to regulations and less interference. [8]

3.1 Wireless for industrial mobile robotics

3.1.1 Architecture

The possibility for a mobile robot to accomplish its goals depends on the availability of an effective communication architecture which allows the connection with the central controller.

A representation of an industrial scenario is shown in 3.1: several mobile robots and driverless platforms are deployed within an indoor environment like, for instance, a production facility or a warehouse. These platforms could be one of the AGVs previously described and they may be equipped with one or more manipulator arms to perform additional tasks as picking and placing objects. Each platform is equipped with a battery for power supply, a computational board and a wireless communication interface. When multiple platforms operate together in a coordinated way, they are often termed as "fleet".

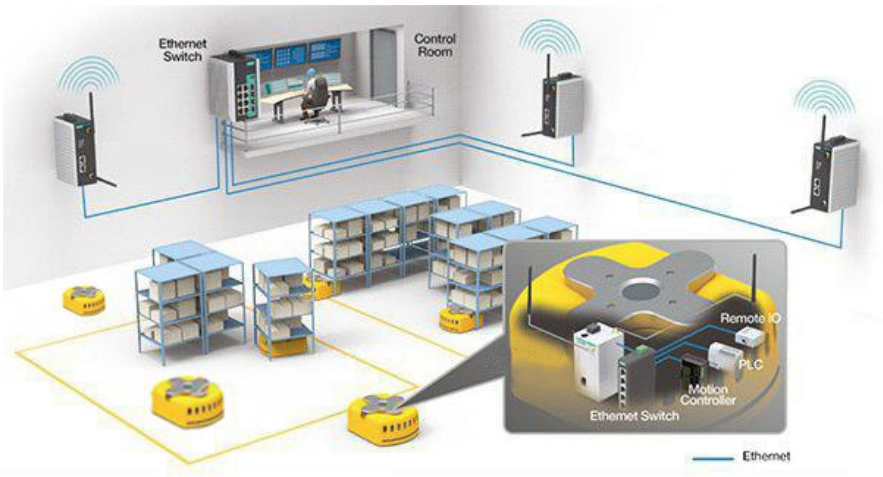


Figure 3.1: Example of a possible industrial scenario. [9]

Depending on the control architecture in use, some of the tasks performed by AGVs in an environment like the one shown in Fig. 3.1 can be carried out by a central controller, located in the same building. The controller is wired connected to access points (APs) which communicate with the mobile platforms through a wireless network. In most applications, the controller is also equipped with an Human-Machine Interface (HMI) and connected to an Enterprise Resource Planning (ERP) or Warehouse Management System (WMS).

It must be noted that wireless communication is needed in every mobile robots application

since their mobile nature prevents the usage of cables. Consequently, the choice of the control architecture strongly impacts the communication performance that the wireless network must provide.

Two main paradigms are possible: distributed and centralized control. In the former, the brain which elaborates the tasks is located in the robot itself, equipped with strong computational power and a large set of sensors. In this case, the central controller will simply transmit high-level information and possibly receive back some status information. Based on the received information, the mobile device will autonomously execute path planning, mapping, localization, obstacle avoidance and all the other required tasks. The data exchange over wireless can be made asynchronously at low frequency.
In the centralized control paradigm, the mobile device is equipped with a small amount of computational power and sends all or most of the data from its local sensors to the central controller. The controller processes the received data and replies with low-level control commands, ready to use for the platform to execute the next tasks. Here, the exchange of data over wireless follows a synchronous pattern with very high update rates.

Hybrid approaches are also possible solutions. For instance, some tasks can be executed locally as obstacle avoidance, whereas other ones are executed by the central controller as path planning and fleet management.

DISTRIBUTED CONTROL	CENTRALIZED CONTROL
Lower communication needs	Lower device cost
Faster reaction to dynamic changes	Easier synchronization and collaboration among different devices

Table 3.1: Advantages of the two possible control architectures.

In Table 3.1 the benefits for the two control architectures are summed up. The centralized approach seems to be better suited for complex logistics applications, with several mobile devices executing different tasks simultaneously. Indeed, such an approach allows to easily coordinate and synchronize several devices and, moreover, the controller can establish a general view of the plant, allowing holistic optimization to be performed.

The choice of the system architecture is hence an important problem. The centralization makes some aspects of the control much simpler. However, the system has a single point of

failure and when a fleet with a significant number of AGVs is involved, the network is heavily loaded. On the other hand, a distributed solution relieves the load of network, but it may require the use of peer-to-peer communication links.

It must be remarked that fast reaction to dynamic changes is a requirement regardless of the chosen paradigm to guarantee personnel safety, especially when machines and human workers share the same environment. Hence it is essential to deploy wireless networks able to guarantee reliable data exchange at high update rates.

3.1.2 Available wireless solutions

In this section, all the best wireless technologies available on the market providing an high transmission rate are discussed.

Wi-Fi

The IEEE802.11 is a part of the IEEE802 family of LAN protocols. It specifies the two lowest layers of the ISO/OSI stack, namely, media access control (MAC) and physical layer (PHY), to deploy wireless local area networks (WLAN), which are better known as Wi-Fi. It mainly exploits unlicensed frequencies as 2.4 and 5 GHz and nowadays it is the world's most used wireless computer networking standard.

AMENDMENT	RELEASE DATA	BANDS	PEAK RATE
802.11n	2009	2.4/5 GHz	600 Mbps
802.11ac	2013	5 GHz	6.77 Gbps
802.11ad	2012	60 GHz	6.76 Gbps
802.11ah	2017	900 MHz	234 Mbps
802.11ax	2020 (expected)	2.4/5 GHz	9.61 Gbps
802.11ay	2019 (expected)	60 GHz	176 Gbps

Table 3.2: Relevant IEEE802.11 amendments.

The first IEEE802.11 standard was published in 1997 and after that several amendments have been released in order to improve its capabilities. Table 3.2 summarizes most relevant amendments showing their main characteristics.

Two other unlicensed bands are currently being explored, the 60 GHz (known as mmWave), which offerss higher throughput, and the 900 MHz one for longer range and lower power consumption. Unfortunately, only a few WiFi chipsets currently support IEEE802.11ad, for 60 GHz bands, and no one supports IEEE802.11ah (for 900 MHz), so almost all WiFi deployments are still in the 2.4 and 5 GHz bands.

Soon, two major amendments should are expected to be released, the IEEE802.11ax and the IEEE802.11ay. The former introduces a new channel access method, Orthogonal Frequency Division Multiple Access (OFDMA), which is already adopted in cellular networks to allow scheduled channel access in time and frequency, the latter is an evolution of IEEE 802.11ad, because its goal is to achieve higher throughput in the mmWave spectrum by means of channel aggregation and MIMO.

One of the reasons why WiFi is the most adopted wireless technology on the world is that it makes possible to achieve significant data rates in comparison with other standards. Indeed, in the latest amendments, multi-Gbps data rates are obtained which are comparable to the currently available Gigabit Ethernet technology. Its main disadvantage, that makes it not well-suited for industrial applications, is that it uses Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) as a channel access mode: each node listens to the channel before transmitting a packet and it backs off for a random amount of time if the channel is busy to avoid collisions. This mechanism is adequate for fair channel access in "best-effort" applications, but on the other hand it does not guarantee punctual data transmission and consequently is not suitable for real-time applications, such as control of AGVs and mobile robots.

Typical industrial applications implement a Time Division Multiple Access (TDMA) technique to access the channel, in this way each node transmits exclusively during a predefined time slot and so they cannot interfere each other. Up to now, no IEEE802.11 amendments practically offer a TDMA implementation over WiFi. The OFDMA mode that will be introduced in IEEE802.11ax might approximate TDMA and offer better support for real-time, but it has not been fully demonstrated yet.

Two interesting features covered by some IEEE802.11 amendments are mesh networking and fast roaming. The former is put into practice by IEEE802.11s, defining routing protocols and

security methods to realize multi-hop communication among IEEE802.11 mesh stations providing better coverage, scalability and robustness. The latter is supported by IEEE802.11r, which aims at seamless handover between WiFi APs, by simplifying the security procedures when a station hops from one AP to another. Experimental studies with an open-source implementation of IEEE802.11r showed that the handover delay can be reduced to 19 ms [10]. Although IEEE802.11s and IEEE802.11r could be useful, they are not implemented by all off-the-shelf WiFi products and might be difficult to configure.

Accounting only what the market provides, the currently available chipsets supporting IEEE 802.11a/g/n/ac can offer the following performance:

- For what concerns latency, reliability and coverage, in [11] is shown that the minimum time for transmitting a 100 bits message with 99,99% reliability via IEEE802.11ac is 126 μs considering all overheads, channel access and acknowledgement, with a link range of 100 meters [11]. It is necessary to highlight that reliability and range are tightly related and can be enhanced through better modulations, coding and retransmissions at the price of increasing the latency.
- Since WiFi adopts a randomized channel access mode, the maximum recommended number of active devices on the same access point is around 25; a greater amount of nodes can be achieved by deploying multiple APs on different channels. [12]
- The support to roaming is a big issue in IEEE802.11. Generally speaking handover delays are larger than 1 second [13]. Only with IEEE802.11r it can be theoretically lowered to 20 ms. [12]

Bluetooth

Bluetooth is a very popular wireless technology used for several low range applications; it was created by Ericsson in 1994, lately standardized as IEEE802.15.1 in 2002 and then handed to the Bluetooth Special Interest Group (SIG) in 2005, that is still managing its evolution. The latest amendments include the Blueooth 4.0, released in 2010, whose main feature is Bluetooth Low-Energy (BLE) for extremely low power consumption. Bluetooth 5.0 was released in 2016 to target even longer range and lower consumption. All Bluetooth standards work in the unlicensed 2.4 GHz band.

Bluetooth's peculiarity is that its transmitters exploit very narrow channels of 1 MHz and jump constantly from one channel to another according to a pseudorandom pattern ("frequency hopping"): this permits to achieve robustness to fading and external interference. The BLE feature allows extremely low power consumption but in general it offers a low data rate and this is a drawback for critical control applications.

Bluetooth deployments offer the following performance:

- Data exchange in Bluetooth is organized in slots of 625 μs and its usage is meant to be for close-proximity applications, hence the reliability drops below 99% farther than 10 meters [14]. In Bluetooth 5.0, increasing the transmission power, a nominal range up to 400 meters is possibly achieved.
- Bluetooth networks, also called piconets, are made at most of 7 devices. More piconets can be deployed in the same environment thanks to frequency-hopping, however when more than 5 piconets are active simultaneously the reliability drops below 95%. [14]
- Roaming from one piconet to another one is generally not supported unless the scatternet protocol is used. However, only few implementations of scatternet have been realized due to limitations of Bluetooth.

EchoRing

EchoRing [15] is a wireless token-passing protocol, released in 2015, to support factory automation and the smart grid phase synchronization. To carry out communication with low latencies and high reliabilities as well, this decentralized protocol introduces the concepts of cooperative communication and an improved fault tolerance.

The authors of this protocol specifically designed a Data Link Layer (DLL) as a custom token ring based medium access scheme to deploy on top of the IEEE802.11 PHY. In fact, token-passing permits strict latency and deterministic medium access, but maintaining its stability over wireless links is rather challenging. Hence the classical scheme is unsuitable for wireless industrial networks.

This mechanism is based on granting the access to the medium only when a station owns

an exclusive transmission right, called token. This right is forwarded along all the stations and whenever it is received it must be acknowledged. Given the broadcast nature of the wireless medium, a station could be connected to all the other ones, but a logical ring topology is imposed and therefore every node has got a predecessor and a successor.
When a station has the token, it can hold it an amount of time called Token Holding Time (THT), during which a station can transmit every sort of packet.

Station failures and transmission errors are the two main problems that can lead to a ring instability or payload packet losses. Automatic Repeat reQuest (ARQ) can be employed to correct token transmission errors, but retransmissions may still fail because they are constrained by the THT. After a specified number of unsuccessful retransmissions, the station with the token excludes the unresponsive successor and tries to forward the token to the successor of the unresponsive station; in case of persistent failures this may lead to the ring disintegration. The other problem occurs when the token disappears. The station waiting for the token creates a new token and, after a while, a new rotation cycle is started adding significant latency to data transmission.

When dealing with low latencies, it is not possible to exploit time diversity, therefore the only way is to employ spatial diversity. The solution for the first of the two problem is cooperative ARQ which allows to select another station to assist during the transmission process: in case of loss of an acknowledgment, the support station will echo it. Cooperating stations have to be determined, pushing stations to get Channel Quality Information (CQI) expressed as Signal-to-Noise Ratio (SNR) for as many nodes as possible. Every station stores a local connectivity matrix containing the SNR estimation for each directional connection in the ring. All stations share the information regarding their links into the token and other stations will use this information to update their local connectivity matrix. At most one cooperating station is chosen and this is enough to guarantee spatial diversity. This is a key feature of EchoRing.

For what concerns the second issue, a station can benefit from the broadcast nature of the medium by overhearing transmissions between other stations. After a burst of unsuccessful retransmissions, one station could assume the link with the successor as permanently broken and consequently excluding the related station from the ring. Often, it happens that a link may become stable again after some cycles. Hence, a station may retrieve information

regarding that link later on analyzing the overheard packets and coming back to the normal situation. A lot of ring instabilities and ring disintegration can be avoided thanks to this solution.

The performance of this protocol are detailed in the following:

- Considering the coverage, since this protocol lays over the 802.11 PHY layer, it can span theoretically around 100 meters. Anyway, in [15] it is suggested to operate in a shorter range like 15 meters.
- Number of nodes, latency and reliability: the performance of these three parameters are tightly related that can been seen as trade-off. For example, if a latency of 2 ms and a reliability of 99.9999% are demanded, only 2 stations are possible. If we loose the latency to 10 ms with the same reliability, the number of stations increases to 15. [16]
- Roaming support: the feature in which a station can migrate to another ring is not conceived.

Although commercial devices are already on the market by R3Communication, this solution does not cover all the the use cases shown in 2.1, in particular when the update rate required is stringent. However it provides a good solution for non-rigorous scenarios, for instance with 20 ms minimum latency and a reliability of 99.99%, a fleet of 44 AGVs can be controlled.

LTE

Cellular communication systems are one of the most popular examples of wireless networks. They were introduced for personal use but now they are targeting also the industrial scenario.

Their standardization has been handled over the years by organizations different from IEEE and versions are known as “generations”. The evolution of cellular networks has been handled by 3GPP since 2003 with the release of the third generation (3G), then the fourth generation (4G), also known as LTE, was released in the end of 2010. Even though LTE is providing some solutions for industrial applications, the fifth generation (5G), that will be released in 2020, is expected to also target machine-type communication for industrial use cases.

As already anticipated, cellular networks use proprietary spectrum, which is auctioned by government to the various telecom operators. However, the trend of local LTE networks that can be deployed by private entities, using dedicated radio equipment must be taken in

consideration as feasible solution. These private networks can use either unlicensed or licensed spectrum.

For what concerns the unlicensed scenario, with the foundation of the MulteFire alliance in 2015 which includes the major communication players, it became easier to deploy private LTE networks in the 5 GHz band without any relation with operators. Conversely, for the licensed spectrum, the Citizens Broadband Radio Service (CBRS) Alliance in USA makes available to privates in the 3.5 GHz band small slices of 150 MHz following two different licenses:

- priority access, auctioned by the government to users for a limited amount of time;
- general authorized access, allowed to every user under strict regulations.

In Europe, only Germany announced that some of the frequency bands for 5G will be licensed to private enterprises. [17]

The creation of such private local networks in the unlicensed spectrum is interesting for industrial communications because they employ OFDMA as channel access method, even if they lose the most important advantages, that is the exclusive use of the medium. Therefore this scenario allows better determinism by means of the complex scheduling and high level of scalability, supporting more than thousands of nodes in the same network. Furthermore, since this kind of network was designed for user equipment moving with random patterns at high speeds, it provides support for mobility because handover is a key feature in this technology.

Since LTE is a complex technology, the main drawbacks are that its deployment is very expensive and communications are affected by high latency. Building an LTE cell including core network and base stations is obviously more expensive than purchasing equipment for any WiFi and Bluetooth network. The minimum achievable end-to-end latency is now in the order of 50 ms and improving this value is one of the main goals of the upcoming 5G standard. Since transmission power must undergo strict regional regulations which impose low transmission power in the unlicensed bands in general, the coverage area is consequently reduced in unlicensed deployments. However, the range is still longer than WiFi, with the ability of reaching up to 100 meters with maximum throughput and high reliability.

Since the introduction of unlicensed cellular networks is a recent solution, there are no established industrial standards based on this technology.

Considering the private LTE implementations in licensed spectrum, the following performance are expected:

- Latency, reliability and coverage: the LTE superframe is 1 ms but the final end-to-end latency is much higher (around 50 ms), because of the complexity of the channel access and core network delays [18]. LTE range is quite wide, a cell coverage can span from a radius of few meter to many kilometers and this is possible with its high transmission power and typical high position of base stations, guaranteeing a reliability of 99.9%.
- Number of nodes: OFDMA allows LTE to host thousand of devices in a superframe; within a 20 MHz channel, 100 resource blocks are carried in each frame, which can be potentially assigned to 100 different users.
- Roaming support between different cells is one of the key features in LTE and handover delays are in the order of 50 ms.

It is expected that the landscape will further evolve with the release of 5G thanks to the definition of a profile specific for industrial applications, called Ultra Reliable Low Latency Communication (URLLC). This profile promises latency below 1 ms and transmission reliability over 99,999%.

Table 3.3 summarizes the performance of main features of communication protocol described so far.

TECHNOLOGY	FUTURE	LATENCY	COVERAGE	NETWORK SIZE	ROAM
IEEE802.11ac	802.11ax/ay	$\sim$100μs	$\sim$100m	25 nodes/AP	>1 s
Bluetooth 4.0	Bluetooth 5.0	625μs	$\sim$10m	7 nodes/piconet	-
EchoRing	-	20ms	15m	44 nodes	-
LTE-Advanced	5G	$\sim$50ms (<1ms 5G)	>1 km	100 nodes/superframe	$\sim$50ms

Table 3.3: Typical performance of wireless standards.

WIA-FA

The WIA-FA protocol, developed by the Shenyang Institute of Automation (SIA) and standardized as IEC62948 in 2017, aims at supporting factory automation applications on top

of standard WiFi in the 2.4 and 5 GHz bands. Currently, this is the only industrial wireless standardized protocol for factory automation based on WiFi. WIA-FA adopts a customized MAC layer, including TDMA, on top of the IEEE802.11a/g PHY. Even though the maximum data rate is limited to 54 Mbps, the use of TDMA allows deterministic data exchange. Further features for reliable transmission and mobility support have been added. WIA-FA products are distributed by Shenyang Zhongke Allwin Technology Co. Ltd. (ZKAW) and they claim that their networks can support more than 100 nodes with update rates around 100 Hz.

This protocol will be deeply discussed in Chapter 2, since it is the main wireless technology utilized in the following experimental studies.

3.2 Related work

By browsing the literature, no related works to EGM performance were found, mostly because it is an exclusive library for ABB robots. In general, there are very few papers reporting practical deployments of wireless networks used in critical motion control applications. Several solutions are present for bridging CAN with a wireless technology, but for most of the cases no performance results are shown.

3.2.1 Wireless for motion control

An interesting scenario is presented in [19], a research test-bed is built in LuMaMi, a Massive MIMO test-bed at Lund University, to study mission-critical control over a 5G distributed network supported by Ericsson.
It resulted that while performing the task of remotely handling a ball, a latency of 10 ms in a round-trip communication was introduced by the wireless link. Furthermore, from the motion perspective, 5G appeared more reliable than LTE because the process remained stable and was performed without interruptions. Then it is also showed that the offsets between the set-points and real positions are in the order of a few centimeters.
5G is showing good results, but it seems not enough mature for critical motion control. There are high expectations for the future network slice implementation, URLLC, which will be discussed in Section 7.

A Bluetooth motion control application is described in [20] performed at Robotics Laboratory of the University of Szeged. A wireless robot is guided through some obstacles to get a target position and it is pointed out that it is necessary to find a trade-off between avoiding the obstacles and moving towards the target. It is clear that Bluetooth is not suitable for critical motion control.

The last relevant work for the topic is the realization of a real-time wireless (RT-WiFi) solution based on IEEE802.11 with only off-the-shelf products, shown in [21]. The goal of RT-WiFi is to provide real-time high-sampling-rate data transmission.
It is interesting the presented use case in which a mobile assistive robot generating sensing signals at 1 kHz, and an host computer sending control signals at 1 kHz, are able to communicate by means of RT-WiFi, although the high generation rates.
However, since RT-WiFi is based on IEEE802.11, it suffers from highly congested frequency

bands, slow roaming and limited number of nodes.

3.2.2 Wireless-CAN integration

An integration of wireless nodes, connected in meshed Zigbee networks, with other nodes linked together with CANbus is realized in [22].
In the realization, 8 CAN smart sensors are sending 1 message per second and they are forwarded through ZigBee. This scenario is distant from being a similar real-time industrial application.

An interesting solution is developed in [23]. Two separated CAN nodes generate packets with an unspecified rate. These messages are collected by an ARM board which is employed to forward them through Wi-Fi. The architecture adopted is conceptually similar to the one that will be presented in Section 6.
Only the feasibility of architecture is demonstrated and no performance results are shown.

In [24] it shown how is possible to bridge a CANbus communication over Bluetooth. The proposed architecture exploits a series of microcontrollers with a CAN and Bluetooth interface. They are connected via CAN and they act as masters for some Bluetooth nodes. A controller is necessary to trigger the communication. No information are reported regarding data rate and total number of nodes in the realization.
Also here, only the feasibility is demonstrated without resulting performance.

In [25] it is proposed a new collision MAC scheme to offer prioritization and schedulabilty, for integrating CAN with wireless technologies in a real time fashion, called WiDOM. This seems a promising solution, however it was limited to a range of 15 meters.

Gli uomini in universale giudicano più agli occhi che alle mani, perché tocca a vedere a ciascuno, a sentire a pochi. Ognun vede quel che tu pari, pochi sentono quel che tu sei.

Niccolò Machiavelli, il Principe, XVIII

4

WIA-FA overview

Nowadays industrial communication requirements are more stringent than ever for what concerns hard real-time, high reliability and low jitter; therefore many efforts to the development of IWCNs standards have been devoted.

Four IWCN standards have been released by the International Electrotechnical Commission (IEC), namely WirelessHART (IEC 62591) in 2010, WIA-PA (IEC 62601) in 2011, ISA100.11a (IEC 62743) in 2014 and finally Wireless networks for Industrial Automation-Factory Automation (WIA-FA), (IEC 62948) in 2017

We can split them in two categories based on their application. The former three were developed for process automation, while the latter, for factory automation. It is important to point out that WIA-FA is the only approved international standard of IWCNs for factory automation by IEC. [26]

4.1 The WIA-FA system

4.1.1 Devices

WIA-FA introduces a set of physical devices, all of them with specific functions, defined as host computer, gateway device, access device, field device, and handheld device:

- *Host Computer*: it is used by the administrator or the personnel for configuration, monitoring and control in the WIA-FA environment.
- *Gateway Device*: it is the device that acts as intermediary between the WIA-FA network and other networks; it runs data mapping functions and protocol conversions. Indeed, when a packet is received by the gateway, the ISO/OSI header is stripped and replaced with WIA-FA standard header; then the packet is forwarded to the access device. In a WIA-FA network more gateway devices are allowed, but only one is working, called the primary gateway, whereas the remaining are accounted as backup devices.
- *Access Device*: it is the device that takes care of forwarding data between the gateway device and the field devices. The data includes the gateway control command to select the field devices that need to receive the message and the payload. The connection between access devices and gateway device is wired. Conversely, access devices and field devices are wirelessly connected.
- *Field Device*: it is the device attached to a sensor or actuator in the industrial environment. Field devices are allowed to send field data and alarms to other devices, as well as receive configuration information, management information and commands.
- *Handheld Device*: it is a portable device used for provisioning, firmware upgrading, and device state monitoring by directly connecting to the devices using wired maintenance ports.

4.1.2 Topology and Management

WIA-FA uses a redundant star topology, an example is shown in Fig. 4.1. A first star topology is formed because to one gateway multiple access devices are connected, where each of them, in turn, is connected to several field devices creating another star topology.

An access devices cannot communicate with other access devices directly, hence it is up to

the gateway to manage and synchronize them. It is noteworthy that the same address is assigned to the access devices and, consequently, a field device may connect with multiple access devices.

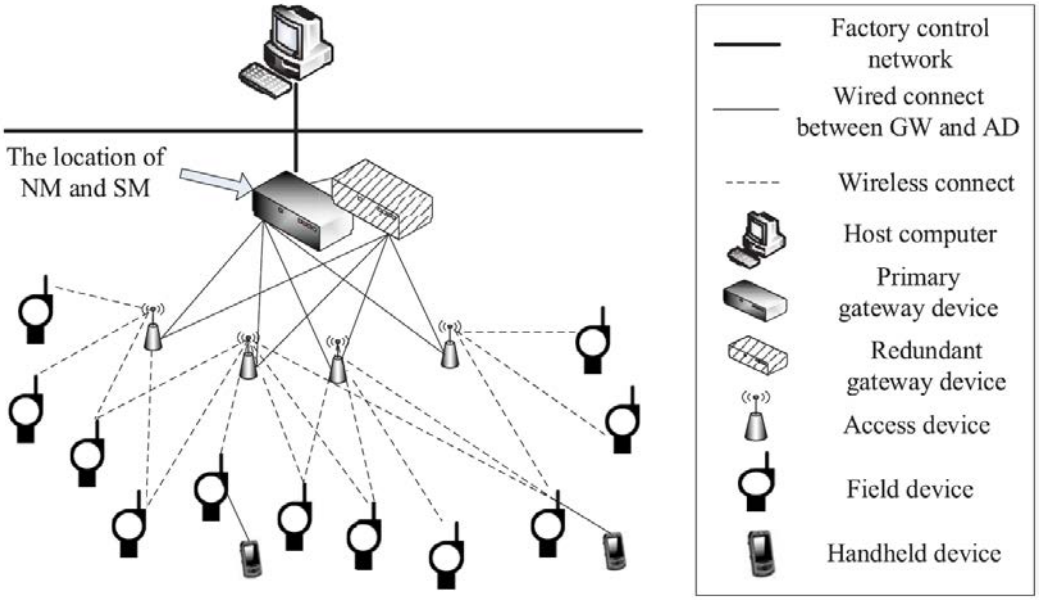


Figure 4.1: WIA-FA devices are deployed in a star topology. [26]

WIA-FA exploits a centralized management framework implemented by the network manager (NM) and security manager (SM) in the gateway device, that are responsible for the management of access devices and field devices. In field devices and access devices instead, network management modules (NNM) and security management modules (SMM) are implemented which perform management functions together with the gateway device.

4.1.3 Protocol stack

The WIA-FA network protocol stack is designed following the ISO/IEC 7498-1 open system interconnection (OSI) model. It defines the physical layer, the data-link layer (DLL) and application layer (AL). Fig. 4.2 shows the relation between WIA-FA architecture and OSI model: it can be seen that the DLL takes care also of the network and transport layer, then the higher layers are considered application layer.

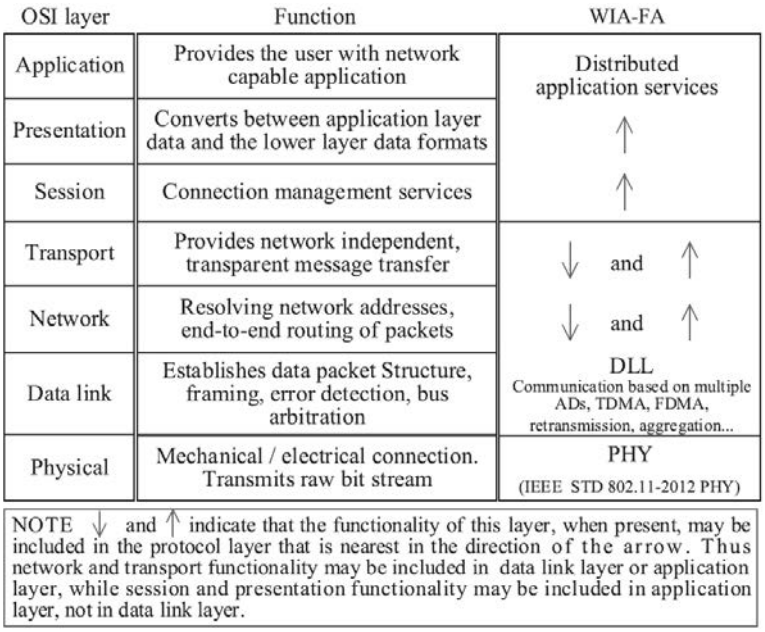


Figure 4.2: WIA-FA stack in comparison with ISO/OSI stack. [26]

Physical layer

The physical layer is based on the IEEE STD 802.11-2012 PHY. WIA-FA supports different modulation modes (FHSS, DSSS, OFDM, etc.) in IEEE STD 802.11-2012 and operates in the license-free 2.4-GHz band. There are three nonoverlapping channels, usually 1, 7, and 11 of IEEE STD 802.11-2012 PHY. The maximum transmit power level should be set according to the regional regulations. The main WIA-FA PHY parameters are summarized in Table 4.1.

Frequency Band	2.4GHz
Number of Available Channels	3
Channel Index	1,7,11
Channel Bandwidth	20MHz
Number of Antennas	1
Modulation Mode	DSSS/FHSS/OFDM/CCK/PBCC
Maximum Rate	54Mbps (DSSS-OFDM)

Table 4.1: WIA-FA Physical layer parameters. [26]

Data Link Layer

The DLL plays an important role in the architecture to implement deterministic communication, it must guarantee real-time, reliable and secure communication among field and

access devices.

WIA-FA DLL adopts time division multiple access (TDMA), this strategy involves superframe to avoid transmission collisions and to obtain, subsequently, reliability and real-time performance of transmission as well as supporting frame aggregation and disaggregation. Furthermore the DLL includes management functions to coordinate devices in the acts of joining, leaving and time synchronization, remote attribute etc.

As the TDMA name suggests, communication is timeslotted, where a timeslot is the time unit for transmitting and its length can be set by the network manager. The gateway generates superframes which are a collection of timeslots repeating at a constant rate cyclically. The length of a superframe depends on the number of timeslots and is also configurable.
A superframe example is shown in Fig. 4.3. It consists of beacon, uplink and downlink timeslots. The function of these timeslot is now explained:

- Access devices exploit beacon timeslots to broadcast their beacon frames in order to let field device joining the network.
- Field devices use uplink shared timeslots to send frames to access devices, this including data frame, join request frames, leave response frames and time synchronization request frames.
- Downlink timeslots are allocated to access devices to send frames to field devices, including data frame, join response frames, leave request frames and time synchronization response frames.

The uplink/downlink timeslot sequence is dynamically configured by the network manager in relation to the necessities of access and field devices. The field devices are informed on the allocated uplink timeslots by the beacons received from the network manager.

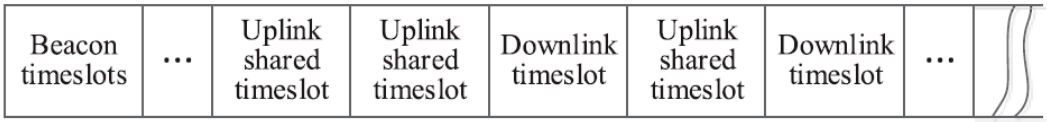


Figure 4.3: WIA-FA superframe structure. [26]

Application Layer

The application layer is composed by the application sublayer (ASL) and the user application process (UAP); the former provides communication services for the latter. UAP can be seen as an unit to implement distributed industrial applications and is made of one or more user application objects (UAOs). As an example, one UAO can be used for the transmission of application data as speed, torque, and acceleration, from industrial processes. UAOs may be located in the same device or different devices. Moreover different UAOs can interact through the ASL.

In addition, WIA-FA defines three communication modes, namely, client/server (C/S) for bidirectional unicast aperiodic non-realtime transmissions, publisher/subscriber (P/S) for unidirectional unicast or broadcast periodic real-time transmissions and report source/sink (R/S) for unidirectional unicast or broadcast aperiodic not urgent transmissions; all of them are meant to satisfy different industrial applications.

4.2 Proprietary WIA-FA technologies

WIA-FA presents several proprietary technologies that make it suitable for different use cases, but not all them are exploited in the experimental applications. Hence in this thesis only the implemented ones will be discussed.

4.2.1 Superframe allocation

Beacon frames are broadcast by the many access devices in the network, so the network manager organizes the access devices into sets, then to each set an available channel is assigned. Access devices in different sets work in parallel in different channels, the idea is that each set does not interfere with each other.

Whenever a field device chooses a channel, it sends a join request in response to a beacon frame, becoming part of the set of the same access device that broadcast the beacon previously. It is necessary to remember that all access devices in the same set are transparent to the field devices, that is if a field device sends a frame, all access devices within the same set will receive the frame.

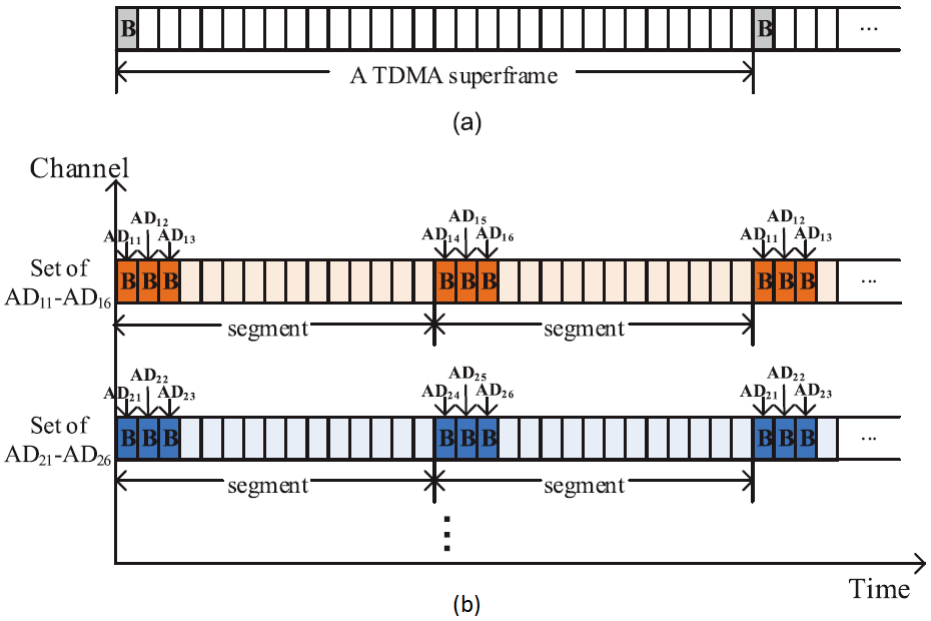


Figure 4.4: Beacon allocation example. [26]

WIA-FA aims to achieve diversity in space and time, so access devices are located in different

positions and the they are divided into teams, with each team broadcasting beacons at different times. The algorithm of team splitting is not specified in the WIA-FA standard.

Two possible examples of allocation of a superframe for only one and, respectively, multiple access devices are shown in Fig. 4.4. In Fig. 4.4a, there is only an access device and the allocation is rather straightforward, the superframe has 30 timeslots with timeslot number 0 is allocated for broadcasting the beacon.

In Fig. 4.4b an environment with twelve access devices is presented, they are split in two sets operating on different channels: the first from AD_{11} to AD_{16} and the second from AD_{21} to AD_{26}. In turn, every set is split in two teams. For instance, in the first set, AD_{11}, AD_{12} and AD_{13} are in one team and AD_{14}, AD_{15} and AD_{16} are in the other. The superframe is divided into two segments (since there are two teams) and the first three timeslots in each segment are used for broadcasting beacon frames.

For what concerns the other kind of frames, the field device is allocated one or even more timeslots for the reception and the transmission of frames with the gateway. Since all the access devices have the same address, the gateway device might receive the same frames multiple times, hence the gateway will drop duplicates. The network manager selects exactly some access devices to transmit to specific field devices, allocating the timeslot for the application. This strategy is put into practice in order to avoid simultaneous transmission of the same frame by multiple access devices. However, the algorithm of selecting access devices by the network manager is not defined in the WIA-FA standard.

4.2.2 Retransmission

Although WIA-FA is a deterministic wireless protocol which guarantees no collision of the transmitted data, it might happen that some packets get lost, damaged or delayed during the communications. In fact, in an harsh industrial environment there are high chances that interference can occur lowering the Signal to Noise Ratio. So taken in account this intrinsic weakness of the wireless channel, WIA-FA was conceived to support four retransmission modes, but only the implemented one will be discussed here.

GACK-Based Timeslot Backoff Mode

The Grouped Acknowledgment (GACK) is conceived for aperiodic frames and uses contention-based access. After the gateway has received aperiodic data or management frames from multiple field devices, it produces a GACK frame according to the addresses of these field devices and it is broadcast multiple times.

If a field device does not receive a GACK frame or the received GACK frame does not include its address, the field device will retry the related aperiodic data or management frame by using the timeslot backoff method to compete retransmission timeslots.

Some timeslots in each superframe are used to retransmit the frames and they are called shared uplink timeslots, because all field devices that have experienced a failure must compete with the others to gain the opportunity to send the data. If a field device fails in the competition it is backed off for a certain amount of time, so it will delay its retransmission to next retransmission timeslot until *MaxRetry*.
An example of the GACK-based timeslot backoff mode is shown in Fig. 4.5.

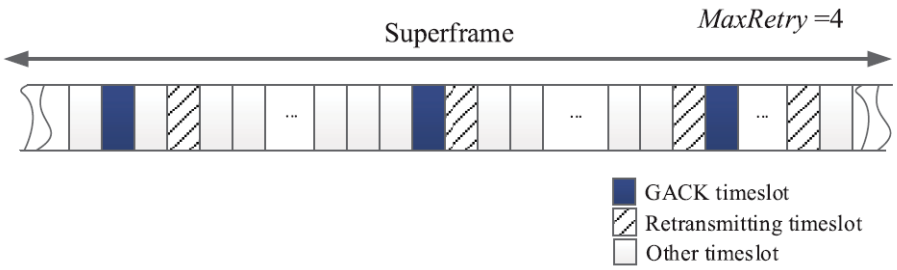


Figure 4.5: GACK retransmission mode example. [26]

4.2.3 Frame aggregation

If an access device needs to sends more frames to a specific field device, these frames can be merged in a unique frame, called aggregated frame. This feature is put into practice by the DLL and in this way WIA-FA reduces the number of transmissions. Moreover WIA-FA DLL takes care of disaggregation of the aggregated frames in field devices.

It is necessary to say that these two mechanisms are optional and it is indicated by means of a flag, *AGGSupportFlag*; when it is 0, the mechanism is not supported in the network.

Then, another flag, *AGGEnableFlag*, is used if the aggregation is enabled (1) or not (0). These flags must be configured in the management information base of the access and of the field devices and it can be applied only to periodic data with the same priority. If the length of the aggregated frame payload exceeds the maximum length of the DLL payload, then the payload will be fragmented.

If *AGGEnableFlag* is set to 1 (which implies *AGGSupportFlag* is 1 as well), the aggregation and disaggregation mechanisms are enabled and the format of an aggregate frame is shown in Table 4.2.

	The first frame			...	The *n*th frame		
1 octet	1/2 octets	2 octets	Vari able	...	1/2 octets	2 octets	Vari able
Aggreg ation number	Field device address	Data length	Data	...	Field device address	Data length	Data

Table 4.2: Payload format of an aggregated frame. [26]

The first octet specifies how many frames are into the aggregated frames, followed by half octet for the short address of the destination field device and by two octets for the data length. In the end, the data to send are appended.

4.3 Practical applications

Two example of real WIA-FA applications are given in [26].

The first example consists of accomplishing a task by means of the synchronized control of collaborating AGVs. The master AGV is equipped with two wireless modules: one for the communication with the scheduling server and the other for the communication with 15 slave AGVs. Each slave AGV is equipped with one module for the communication with the master. All the motion information of the slave which include location, velocity, acceleration, etc. are reported by their master to the server periodically. When a group of AGVs is assigned with a task, the AGV server sends the scheduling information about the task to the master AGV which, then, breaks down the task in subtasks and generates the motion control for each AGV slave. Subsequently, the master passes the motion control instructions to the slaves.

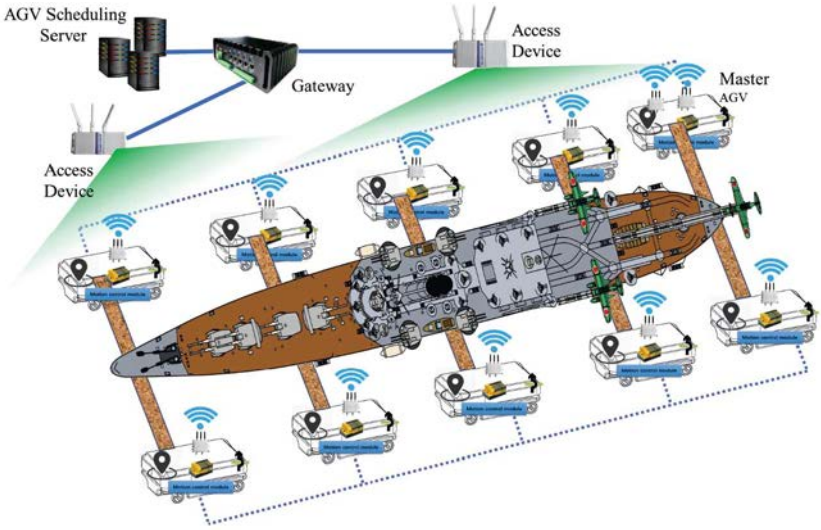


Figure 4.6: Example of AVGs synchronization executing the coordinated task of moving the cockpit of a boat. [26]

In a simple lab implementation with 16 AGVs moving at 0.5 m/s, it is resulted a reliability of 99.9999%.

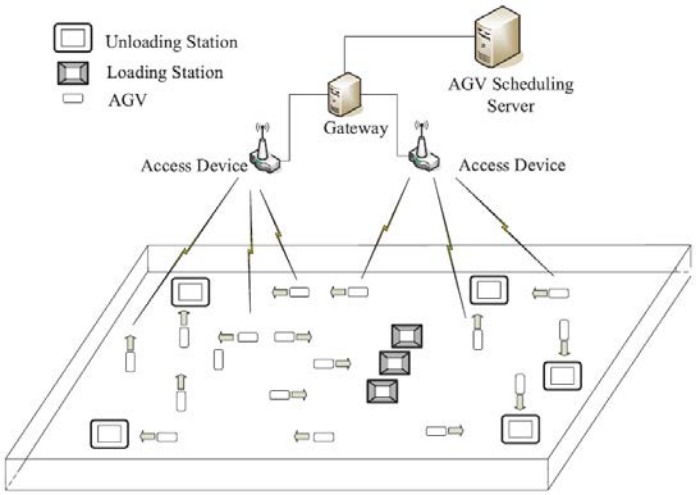


Figure 4.7: AVGs sorting system architecture. [26]

The second example is related to controlling a fleet to provide a logistics sorting system. A real logistic facility of 3200 m^2 was used and it comprehends three loading stations and five unloading stations. On the ceiling, 2 access devices are placed, they are wired connected to the gateway which is in turn connected to the scheduling server. 42 AGVs equipped with a field device are involved to carried out the tasks and they move at 2.5 m/s, as shown in 4.7.

When packages are ready at the loading stations, the server sends commands to the AGVs. The AGVs move the objects from loading stations to unloading stations and after that they return in the waiting area. The AGVs constantly send their locations to the scheduling server which generates and transmits the motion control instructions. The experimental results are that the transmission reliability of the WIA-FA network is 99.99% and the maximum transmission delay is 80 ms.

Dum loquimur fugerit invida aetas: carpe diem, quam minimum credula postero.

Horace, Odes I, II

5 Wireless EGM

Externally Guided Motion is an interface created by ABB to control robots in a low-level manner and it is defined in [6]. This interface allows to control the path of an ABB robot from an external device with a high speed. Indeed, feedback messages regarding the state, position and speed references of the robot are generated with a frequency of 250 Hz.

EGM packets are encoded following the Google protocol buffer structure to ease the data serialization. These packets are exchanged between the client (the ABB robot) and the server (the external device) using UDP as a transport protocol, independently of the physical medium. Ethernet, LTE, WiFi and WIA-FA will be considered to implement the EGM communication.

From the server side, EGM can be installed as a C++ library [27], which needs to be integrated with Robot Operating System (ROS), that is a collection of frameworks to develop robot software.

From client perspective, measurements have been performed on a simulated environment, RobotStudio, and on an ABB robot, YuMi.

5.1 Architecture

The architecture for this experimental use case is depicted in Fig. 5.1.

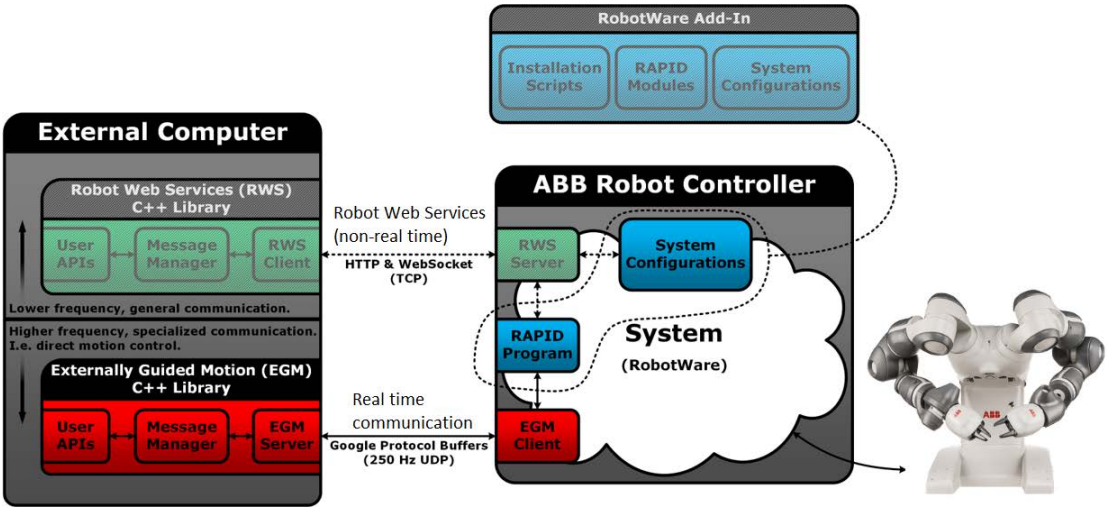


Figure 5.1: Proposed architecture for EGM. [27]

A B&R industrial PC (IPC), that is an industrial-grade computer used in automation environments, represents the external computer. Its operating system is the Linux distribution Debian in which ROS, Robot Web Services (RWS) and EGM libraries are installed. ROS is essential because it allows to use the EGM library.

It is important to highlight that EGM includes two different communication flows. The first one, RWS, operates over HTTP/TCP and it is used to start and stop the interface. In this flow, the external computer acts as an RWS client and the Robot Controller acts as an RWS server. Once the interface is initialized, the real-time communication takes place with the EGM protocol over UDP, in which the external computer acts as a server and the RC as a client.

The RC operating system is running the RobotWare operating system by ABB, that controls the joints of the robot through a Rapid program. The entire RC can be simulated thanks to ABB RobotStudio 6.08 to test movements in a simulated environment in order to avoid critical failures.

The operation begins with the IPC acting as a RWS client, sending a request of initializa-

tion to the RWS server in the RC using the HTTP protocol. This first step can be seen as an handshake between the parties. After that, the EGM communication between the EGM server in the IPC and EGM client in the RC is established. The way to start this process is summarized into a ROS command:

roslaunch abb_yumi_demo abb_trajectory.launch robot_ip:=DST_ADDR_IP

The *abb_yumi_demo* is a predefined trajectory stored in the PC that either the virtual or the real YuMi will follow. The destination IP is the address associated to the RC.

After the connection has been fully established, the EGM Client in the IPC and the EGM Server in the RC are exchanging UDP packets, encoded following Google Protocol Buffer rules. The EGM packet length is 478 Byte and its acknowledgement is 272 Bytes.

5.2 Setups

First, preliminary assessments were performed on RobotStudio using Ethernet and WIA-FA connections. Then, two WiFi commercial modules are used to give a further analysis of the reliability of EGM over IEEE802.11n. Particularly, for a RobotStudio simulation are assessed:

- 1) the position error for Ethernet and WIA-FA;
- 2) the number of packets lost and the length of a round trip transmission for WiFi.

Then, the tests are performed on the ABB robot, YuMi, using Ethernet, LTE and WIA-FA. Specifically, what have been checked are:

- 1) the position error for Ethernet, LTE and WIA-FA;
- 2) the length of a round trip transmission for different WIA-FA timeslots;
- 3) the number of packets lost and the length of a round trip transmission for all the technologies involved.

The involved setups are now discussed.

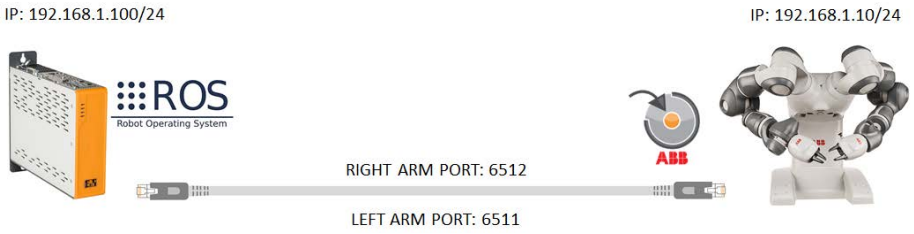


Figure 5.2: Ethernet based EGM implementation.

The setup for Ethernet implementation is shown in Fig. 5.2: the IPC and the ABB robot controller are directly connected with an Ethernet cable. The assigned IP addresses are, respectively, 192.168.1.100/24 and 192.168.1.10/24. A different UDP port is assigned to each arm, in particular 6511 for the left and 6512 for the right.
The WiFi communications is realized using a normal laptop, instead of the IPC, with Ubuntu 16.04, and another laptop running Windows 10. In the former, ROS, EGM libraries and Hostapd are installed; Hostapd is a daemon that allows to use the WiFi module as access point. In the latter, RobotStudio is installed. The connection is established using protocol 802.11 n at 2.4 GHz and channel 6 is employed because it is identified as the least occupied

Figure 5.3: Commercial WiFi based EGM implementation.

after a preliminary spectrum monitoring. The setup is depicted in Fig. 5.3.

A LTE-based EGM implementation is presented in in 5.4. This scenario was put into practice thanks to the Ericsson LTE eNodeB deployed in ABB Corporate Research communication laboratory.

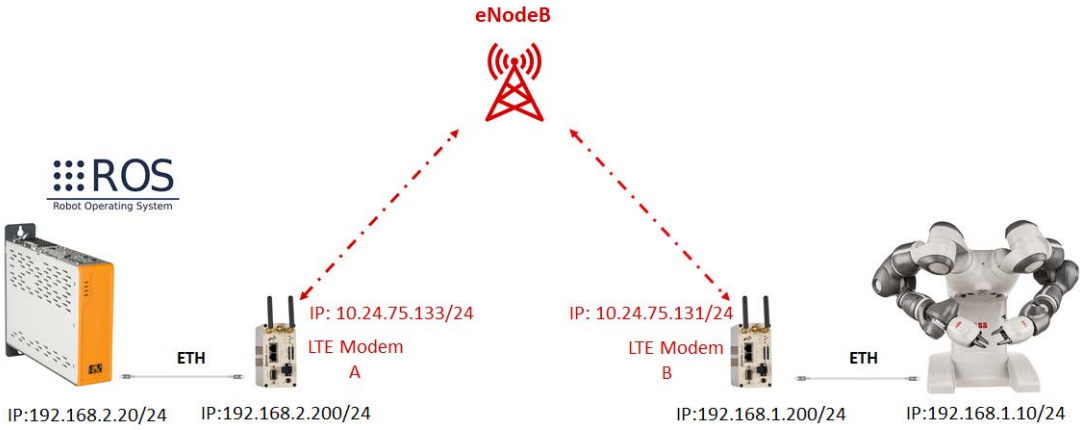


Figure 5.4: LTE based EGM implementation.

The IPC, with IP address 192.168.2.20/24, and the ABB robot controller, with IP address 192.168.1.10/24, are connected via Ethernet to LTE modems. The IP addresses for the IPC modem A are 192.168.2.200/24, for the Ethernet interface, and 10.24.75.133/24, for the one towards the eNobeB. Similarly, for the controller modem B the addresses are 192.168.2.200/24, for the Ethernet interface, and 10.24.75.133/24 for the one towards the eNobeB. The UDP ports 6511 and 6512 are assigned to the left and right arms, respectively. EGM packets are sent from the IPC to LTE modem A, forwarded to LTE Modem B through the eNodeB and then transmitted to Yumi (and viceversa).

The WIA-FA setup uses the WIA-FA devices provided by ZKAW as depicted in Fig. 5.5. The IPC, with IP address 192.168.1.100/24, is connected to an Ethernet switch, which in turn is connected to the gateway and the access device. On the other side, the controller, with IP address 192.168.1.205/24, is not connected directly to the field device, a laptop is placed in between. The laptop has IP addresses 192.168.1.200/24 and 192.168.1.202/24; the UDP ports 6530 and 6531 are used by RC for the left and right arms, respectively, then they are remapped by the laptop to 6511, for the left, and to 6512, for the right. The reasons behind the presence of this laptop are discussed in Section 5.2.1.

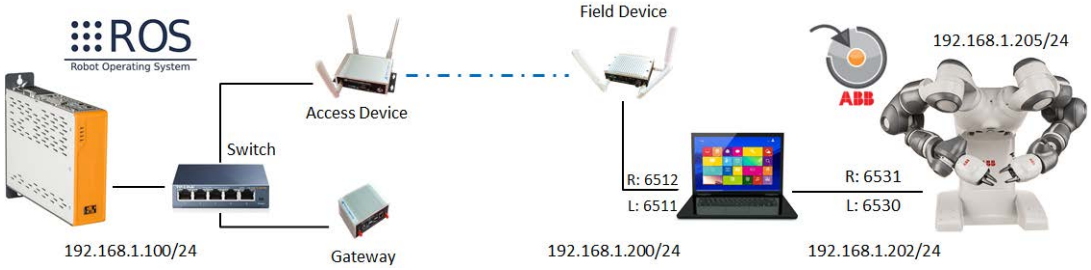
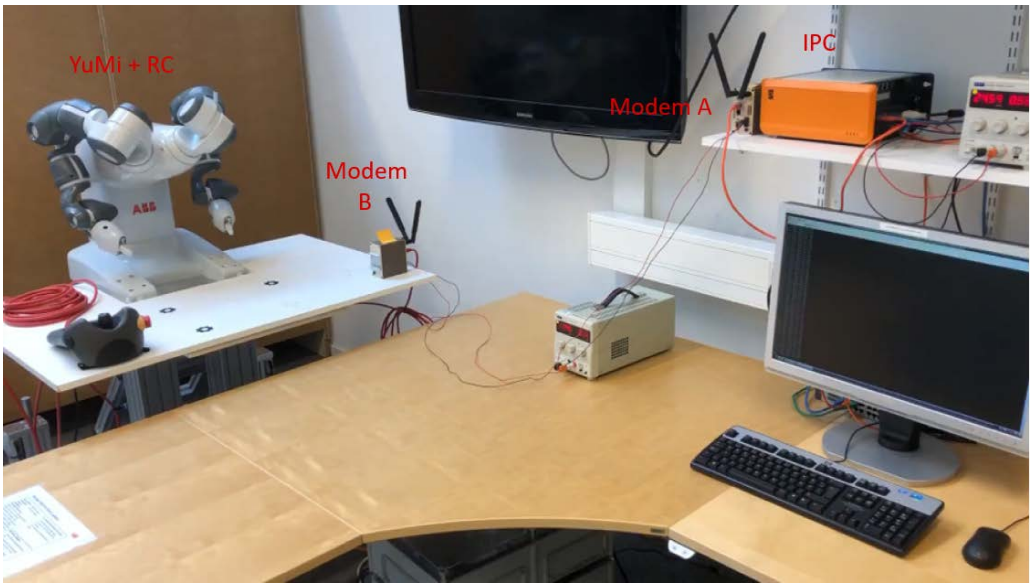


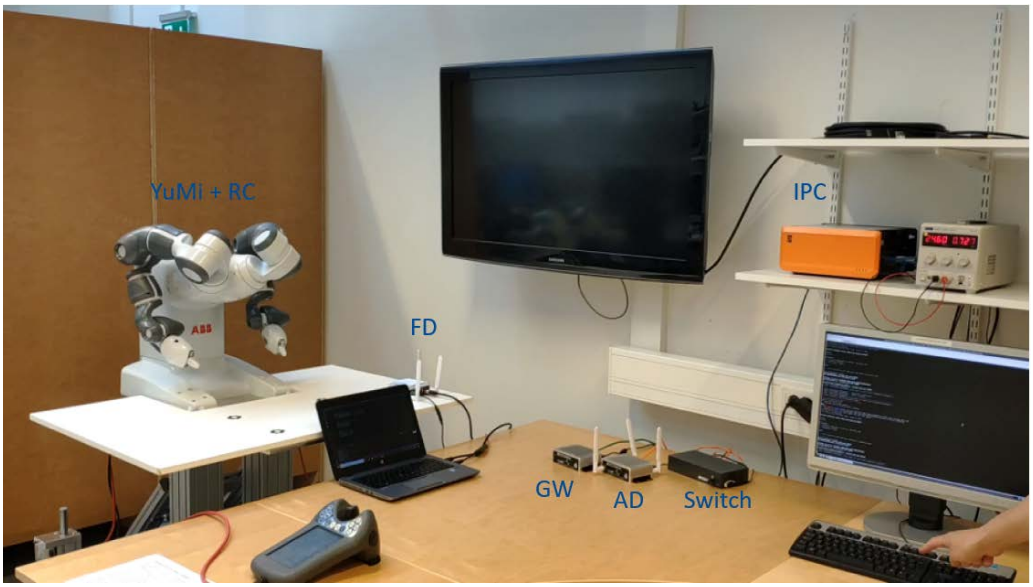
Figure 5.5: WIA-FA based EGM implementation.

In the ZKAW devices for WIA-FA, there is the possibility to configure the slot time length and the protocols to adopt for data communication (i.e. UDP, TCP or only layer 2). The provided setup implements GACK-based retransmission, enabled aggregation but no prioritization and they cannot be changed.

In Fig. 5.6, the lab setup for LTE and WIA-FA are shown. It is important to highlight that, in all the tests, the distance between the IPC and the RC was very small, in the order of a few meters and the communication was line-of-sight. A detailed analysis of the communication robustness in realistic factory environments is outside the scope of this thesis and will be the subject of future work.



(a)



(b)

Figure 5.6: LTE (a) and WIA-FA (b) lab setups.

5.2.1 WIA-FA workaround

A first test of the WIA-FA setup revealed that bidirectional communication between the IPC and the robot controller was not possible. It was difficult to discover the issue since the debugging possibilities on the RC are very limited. To circumvent the problem, a the laptop was inserted in the connection, as a broker, to get more control on the traffic at the cost of extra latency. Then a Python script was designed to handle the packet forwarding both for the virtual robot and the real YuMi.

For the RobotStudio case, a virtual Ethernet interface, with address 192.168.1.202/24, was created with Hyper-V. All the packets generated by RobotStudio were redirected to that address on ports 6530, for the left arm, and 6531 for the right. Moreover, RobotStudio listens to the address for incoming packets.

In the Python script, two sockets are internally listening to the virtual interface, so a bidirectional communication with RobotStudio for the each arm is set up. In addition, two WIA-FA sockets, one acting as server and the other as client, are opened for the right arm; in the same way, two more sockets are involved for the left arm. In fact, just only one WIA-FA socket should guarantee a bidirectional UDP communication, but this setup would not work with the existing firmware provided by ZKAW. To sum up, 6 sockets are opened in total in the Python script to establish a proper UDP communication: 4 for WIA-FA and two internal for RobotStudio.

It is important to point out that, in the real case, only the Python script is adopted without using the virtual interface.

5.3 Results

To assess the control performance, an EGM ROS script in the IPC is exploited. It keeps track of several parameters during the process, that are stored in a log file. The two main parameters chosen to assess the performance are:

- Reference (REF) positions for YuMi's joint number 1, expressed in degrees over time, as planned in the sample trajectory;
- Feedback (FB) positions for YuMi's joint number 1, expressed in degrees over time, as recorded by the RC on the YuMi and sent back to the IPC over EGM.

The difference between REF and FB is the position error, which depends on the quality of the connection over which the EGM is implemented. Although REF and FB (and the error) can be computed for both arms, evaluation for only one arm is reported because there are no meaningful differences.

In addition to control performance, the communication performance was also tested by computing, at the robot side, the packet round-trip-time (RTT) and the packet error rate (PER). They are calculated by making simple modifications in the broker Python script.

Whenever a packet is collect by the script in each direction, its internal sequence number is saved locally with the relative arrival time. At the end of the process, for each couple of entries with the same sequence number, RTT is evaluated as the difference between the associated arrival times. If for a certain sequence number there is only entry, then a loss is accounted and a counter is increased.

The PER is calculated as the ratio between the counter and the total number of involved packets in the process.

5.3.1 Performance evaluation with a simulated robot

The control performance for the simulations ran in RobotStudio using Ethernet and WIA-FA are represented in Fig 5.7. It can be seen that the shapes of the error over time are quite similar: the simulation starts after around 7 seconds, a first peak occurs for both and the WIA-FA's is a little higher. After a small little peak, a third one appears for both, but the Ethernet's is a bit irregular.

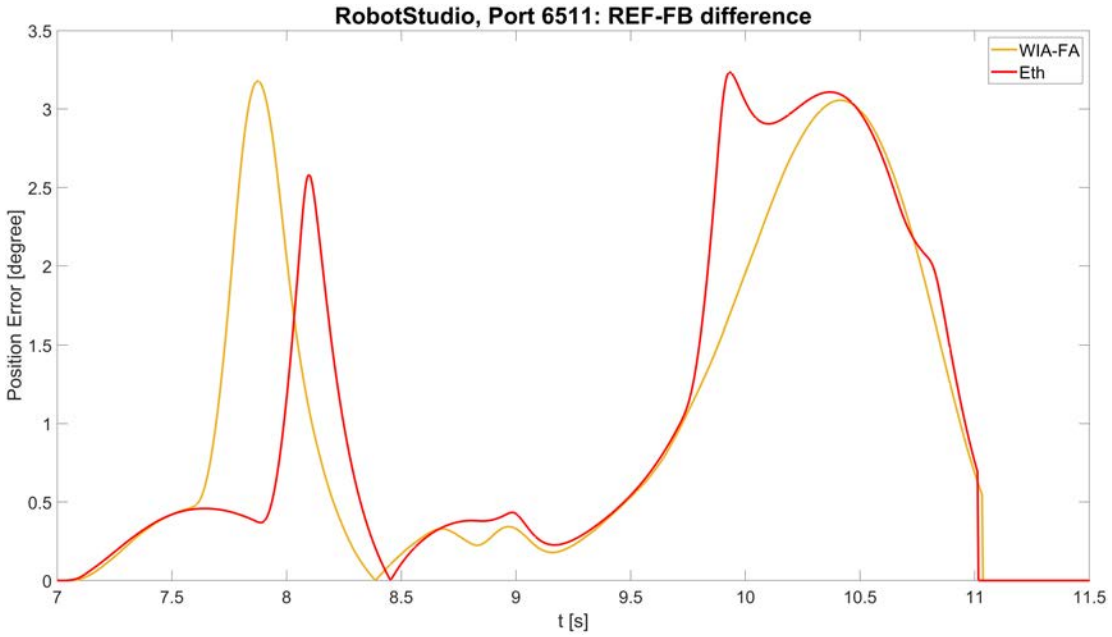


Figure 5.7: Ethernet vs. WIA-FA RobotStudio.

A detailed investigation around the shapes of these position error curves is outside the scope of the thesis and might depend on the selected trajectory. What is important here is that there seems to be no significant difference between the errors with Ethernet (wired) and WIA-FA (wireless). More statistics regarding the simulation are provided in Table 5.1 and, by comparing mean and standard deviation (STD), it seems that WIA-FA performs slightly better.

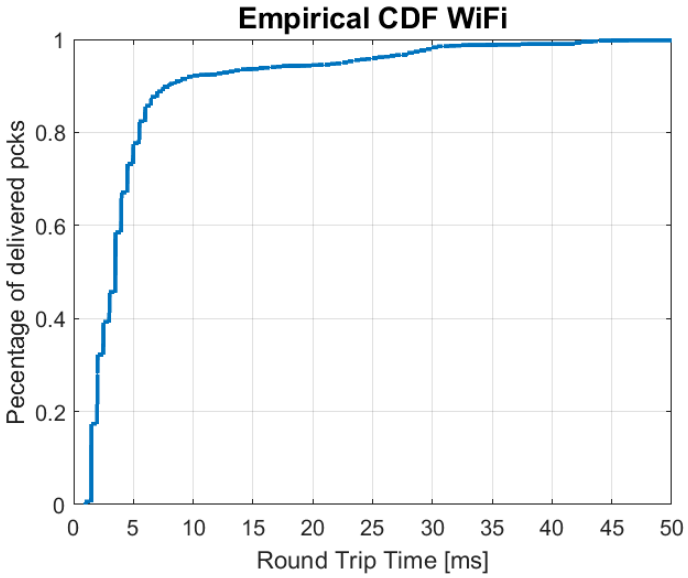
	Ethernet	WIA-FA
Mean	0.3756	0.3668
STD	0.8197	0.7899

Table 5.1: Statistics for RobotStudio comparison - Position Error [degree].

In a further assessment, WiFi was used as communication system. In order to evaluate the

performance, a broker similar to that explained in Section 5.2.1 is used, the main difference is that only two sockets are needed to forward packets through WiFi.

During the analysis, every packet with a RTT higher than 0.07 seconds was considered lost. The test involved 3000 packets.



Mean [ms]	STD [ms]
5.2	0.05

Figure 5.8: Commercial WiFi CDF and statistics.

It is possible to see, in the empirical cumulative distribution function (CDF), in Fig. 5.8, that the 90% of the packets are delivered within a RTT of 10 ms and the average is 5.2 ms. These numbers do not satisfy the necessary requirements and, in addition, between 5 and 15 packets are lost in each test.

These performance are far worse than those offered by Ethernet and WIA-FA, as shown in the following section, confirming the initial intuition that commercial WiFi is not suited for real-time control.

Considered the non-suitability of WiFi for EGM, assessments regarding the position error were not performed for this technology.

5.3.2 Performance evaluation with a real robot

The position error is evaluated again on the real robot, YuMi, for Ethernet and WIA-FA. Moreover, further tests have been carried out with LTE.

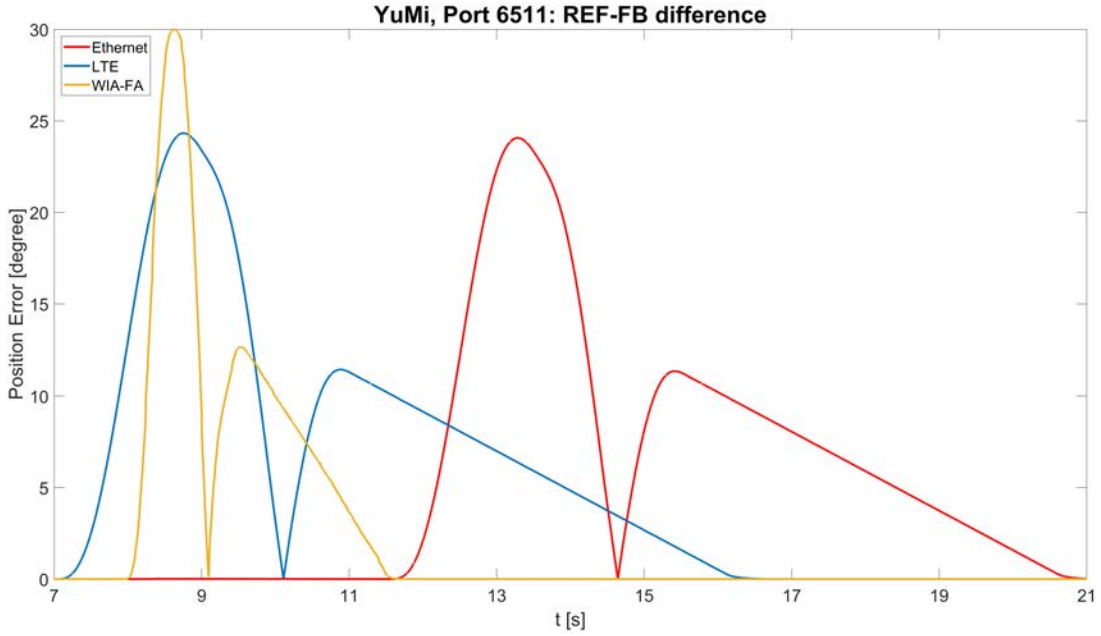


Figure 5.9: Ethernet vs. WIA-FA vs. LTE, YuMi.

It is possible to see in Fig. 5.9 that LTE and Ethernet present almost identical shapes. WIA-FA shape is rather similar but it is more compressed. The movement begins earlier with LTE and WIA-FA than with Ethernet, by around 5 seconds, and this is still unclear. WIA-FA shows an higher first peak than the other technologies, but it is recovered quickly. There is also a second peak that is handled faster by WIA-FA, in fact a steeper slope is present. It must be remembered that it is out of purpose of this thesis to go deeper into the meaning and the position of these shapes.

	Ethernet	LTE	WIA-FA
Mean	1.3664	1.3821	0.6328
STD	4.0976	4.1409	3.1855

Table 5.2: Statistics for YuMi comparison.

The average error for WIA-FA is almost half of the others and also its STD is lower, as shown in Table 5.2. Instead, Ethernet and LTE introduce almost the same error.

Moving to an evaluation of the communication performance, the first step is to optimize the WIA-FA timeslot length. The RTT has been measured for timeslot values of 128, 256 and 512 μs, as allowed by ZKAW devices. Results are provided in Fig. 5.10.

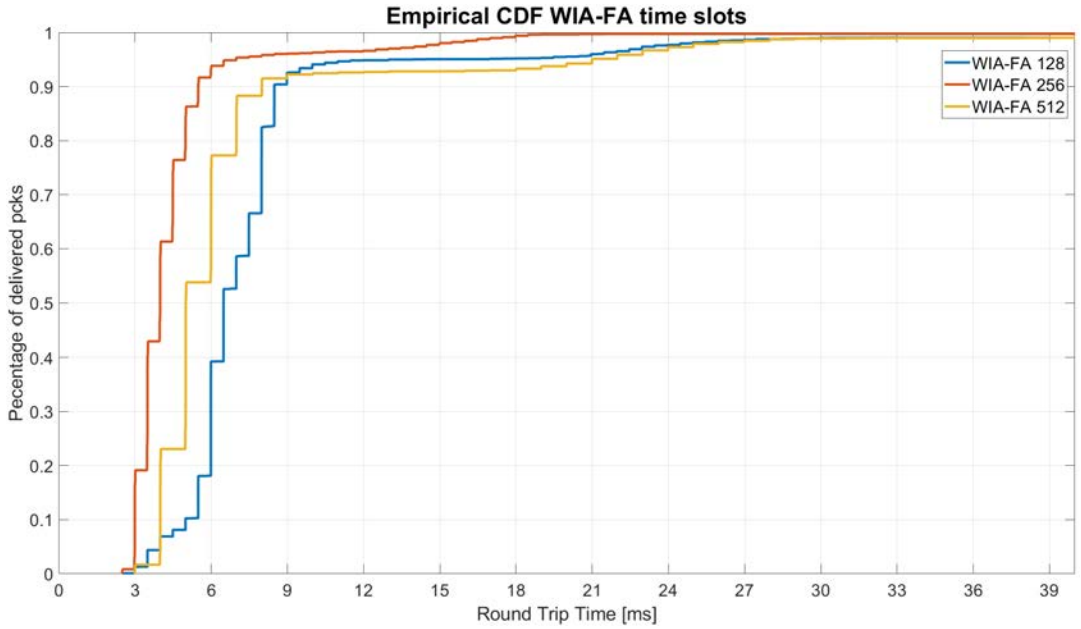


Figure 5.10: Cumulative Distribution Function for WIA-FA timeslots.

From the empirical CDFs, it can been seen that with 256 μs slotlength, the 90% of the packets have completed a RTT correctly in less than 6 ms. This is also confirmed in Table 5.3, showing that the average RTT for WIA-FA with 256 μs slotlength is half of the other ones. Moreover, no packets were lost (PER=0), confirming the reliability of this technology.

RTT	WIA-FA 128	WIA-FA 256	WIA-FA 512
Mean (ms)	8.84	4.87	8.91
Variance (ms)	0.235	0.075	0.267

Table 5.3: Statistics of the RTT for different timeslot.

The slotlength value 256 μs has been selected and from now on, this time slot length will be used.

The RTT has been then evaluated for both Ethernet and LTE. Results are provided in Fig. 5.11 which also reports the CDFs for WIA-FA with timeslots 256 μs.

As can be seen, both WIA-FA and Ethernet behave rather similarly, as the 90% of the packets are delivered withing a RTT around 5 ms. On the contrary, LTE presents a much higher RTT, as expected.

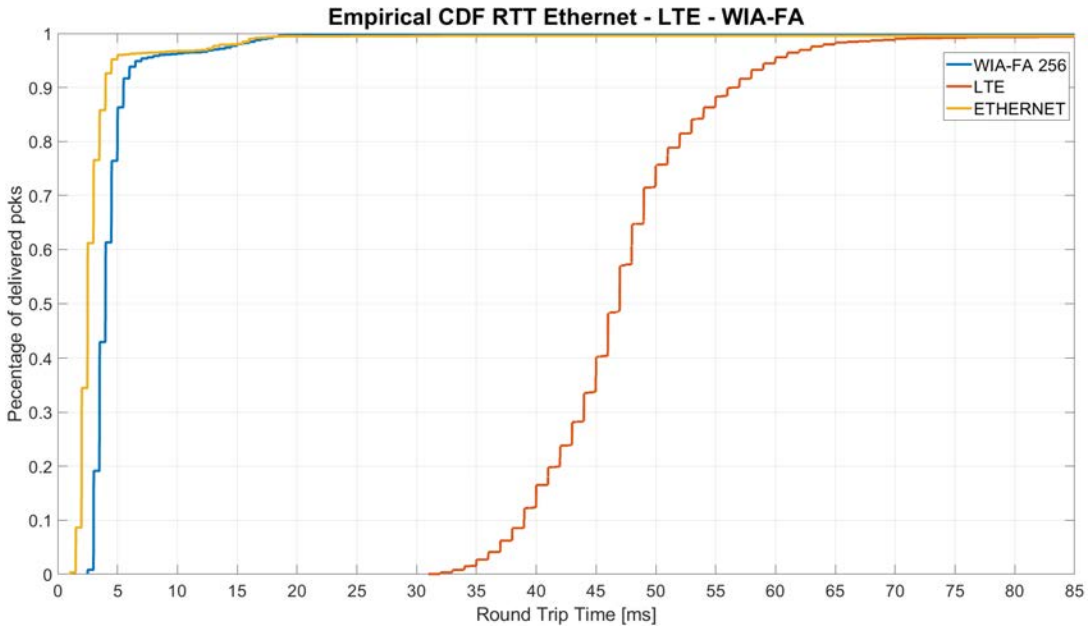


Figure 5.11: EGM RTT, CDF general comparison.

More accurate results can be found in Table 5.4. On average, Ethernet performs around 1 ms better than WIA-FA on average but the latter present a smaller STD.

	Wia-fa 256	LTE	Ethernet
Mean (ms)	4.87	47.94	3.68
STD (ms)	0.075	0.184	0.109

Table 5.4: Statistics of the RTT for EGM communication through WIA-FA, LTE and Ethernet.

5.3.3 WIA-FA layer 2 mode communication

A further investigation was performed by comparing the performance of WIA-FA and Ethernet when exchanging layer 2 packets rather than UDP ones. Indeed, it is expected that communication at layer 2 is more efficient and faster, representing a better choice for a point-to-point real-time communication.

However, since EGM only works with UDP/IP packets, layer 2 communication tests could not be performed using RobotStudio or YuMi. Therefore an alternative way is necessary and a simulated scenario is created as close as possible to a real EGM communication.

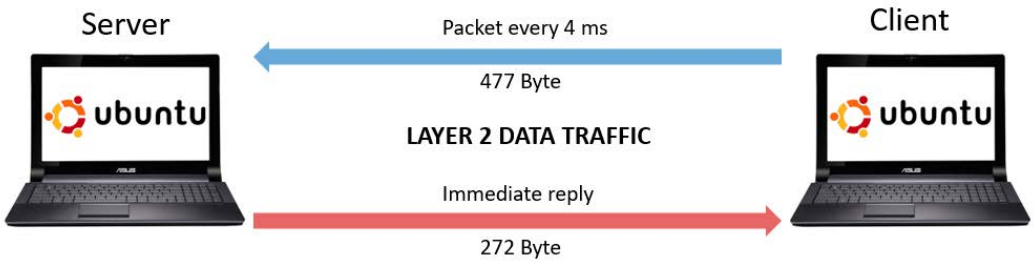


Figure 5.12: Layer 2 mode simulation setup.

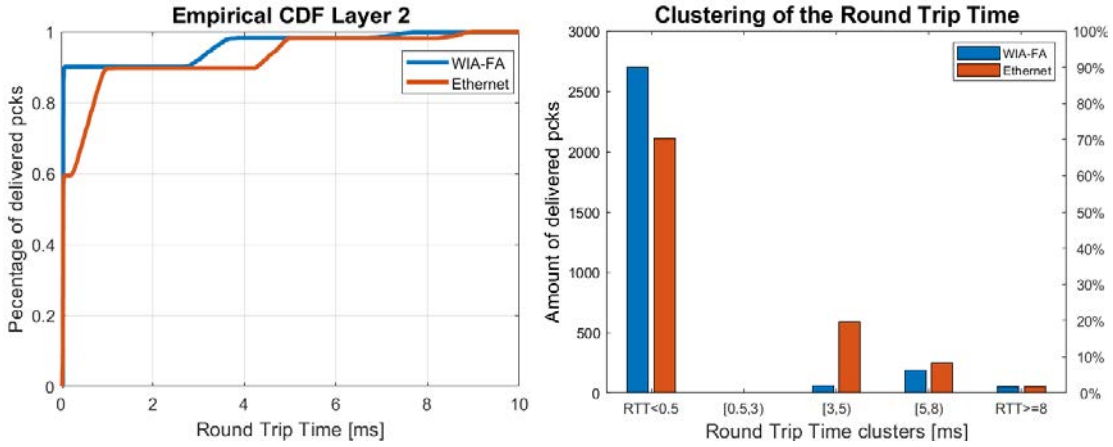
The simulation setup is depicted in Fig. 5.12 and includes:

- Client: an Ubuntu laptop generates every 4 ms a dummy EGM packet of 477 Bytes and sends it through a raw socket;
- Server: an Ubuntu laptop listens for a DLL packet and it replies with a dummy EGM acknowledgment packet of 272 Bytes as soon as a packet arrives.

In the test, a total 3000 packets are exchanged and the results are shown in Fig. 5.13. The empirical CDF presented in Fig. 5.13a demonstrates that the 90% of Ethernet packets are delivered in a RTT around 1 ms. More interesting it is the WIA-FA case, in fact, the 90% of the delivered packets has a RTT lower than 0.5 ms as better depicted more in 5.13b.

The clustering shows that both Ethernet and WIA-FA deliver the majority of the packets with a RTT lower than 0.5 ms, but WIA-FA transmits the 90% of totality against the 70% of Ethernet in the this range.

An overall comparison it is represented in Table 5.5, which confirms that WIA-FA performs



(a) (b)

Figure 5.13: Empirical CDF for WIA-FA and Ethernet (a) and RTT time clustering (b).

	WIA-FA	Ethernet
Mean [ms]	0.45	0.75
STD [ms]	0.002	0.003

Table 5.5: Statistics of the RTT for layer 2 mode.

better than Ethernet.

5.4 Conclusion

After the assessments carried out it can be concluded that WIA-FA supports real-time EGM over wireless, with performance close to Ethernet, both from the communication and control point of views.

LTE and WiFi performance are much lower than those obtained with WIA-FA one, from both latency and determinism perspectives.

Finally, the possibility to send packets exploiting at 2, bypassing the UDP protocol employed in EGM, is appealing to further improve the timeliness of the communication.

The best way to predict the future is to invent it.

Alan Kay

6 Wireless CAN

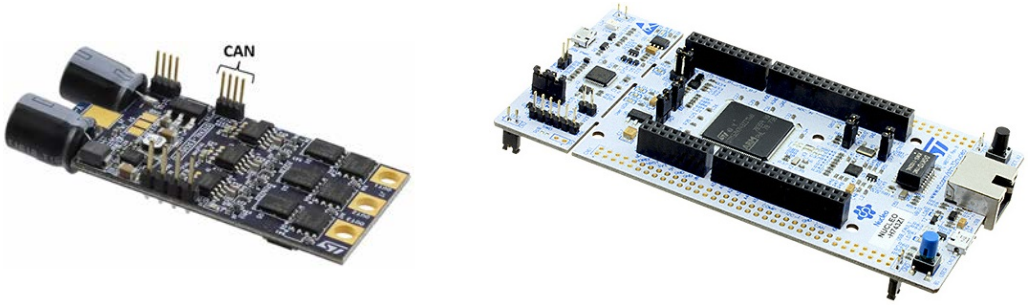
CANbus is a fieldbus used for communication between microcontrollers, which was originally developed for automotive applications. In detail, CANbus is used for the communication between several electronic control units (ECUs) within a vehicle, to carry out several applications such as engine control, transmission, safety systems, etc. Due to its widespread adoption and maturity, CANbus is often the natural choice to implement low-level motion control in mobile platforms, such as AGVs. In these applications, CANbus is typically deployed to connect a central microcontroller to the drive units deployed within each component. Specifically, the application is designed in a master-slave fashion, where the microcontroller (master) sends commands to control the components based on the desired motion set points, while the components (slaves) send feedback status messages.

This thesis explores the possibility of replacing this low-level CANbus communication with wireless links. In detail, rather than connecting each component directly over wireless to the microcontroller, as the first step of this exploration, it has been decided to keep a CANbus among the components and bridge it with the microcontroller through a wireless link. If such an architecture proves to be reliable, new control architecture and applications can be unlocked. For example, the microcontroller can be offloaded from the mobile platform to an external computer, with consequent cost reduction and simplification of the mobile platform. More detailed design of the control architecture is out of the scope of this thesis.

6.1 Setup components

6.1.1 Controllers

For the application considered in this thesis, a master controller is connected to several slave nodes through a CANbus.



(a) (b)

Figure 6.1: (a) F3 controller and (b) F7 controller.

The controller used as slave is the STM32F3 controller from ST, simply referred to as F3 and it is shown in Fig. 6.1a. The F3 is equipped with an integrated CAN transceiver, whose input/output pins are highlighted in Fig. 6.1a. The controller is flashed with a firmware developed in ABB Corporate Research for the drive units of an experimental mobile platform. The firmware generates periodically two types of CANbus packets, one with a fast rate (2 ms) and another with a slower rate (20 ms). The former packet, also defined as critical feedback, contains information regarding the position and state of the motor, while the latter packet includes information regarding voltage and temperature.

The master controller is a STM32F7 device, also from ST, referred to as F7 and shown in Fig. 6.1b. The controller is not equipped with an integrated CAN transceiver, therefore an external one must be connected. Similarly to the F3, the F7 controller is also flashed with a proprietary firmware, periodically generating six kind of CANbus packets each one every 2 ms. In the setup, four packets are used to control different slaves and two (not mandatory) for debugging.

F7 is equipped with an Ethernet interface and it is exploited as debug port. The F7 firwmware is programmed to send UDP packets encoded with Google Protocol Buffer every 0.01 seconds to the IP address 192.168.1.5. A script in C++ integrated in ROS has been developed to

interpret these packets and report each change of state of the slaves or eventual failures.

More information about their circuit schematics and wiring are provided in the Appendix A.

6.1.2 Xilinx System on Chip

In order to bridge the CANbus with the WIA-FA network, it has been decided to use an external board. Indeed, the F3 controllers cannot be connected directly to WIA-FA devices due to a lack of Ethernet ports. For this reason, a System-on-Chip (SoC) board has been selected, namely a XIlinx Zynq-700 SoC ZC706, shown in Fig. 6.2. It is important to remark that the Zynq board does not substitute the controllers in the considered application, instead, it is an additional device exploited to collect CANbus packets and to provide WIA-FA communication.

This board includes an ARM processor and an FPGA: although only the former is used in this thesis, the presence of the latter is an interesting option to implement real-time tasks and can be considered for replacement of both the F3 and F7 controllers in the future.

An ad-hoc linux operative system (OS) designed for ARM processors is run on the platform. Petalinux is the OS adopted because it eases the building of the kernel and the selection of the tools to install. Furthermore, there is no graphical environment. Therefore it provides a light and fast solution.

Figure 6.2: Xilinx Zynq-7000 SoC ZC706.

Another peculiarity of this platform is that it is equipped with several communication ports,

namely USB, UART and two Ethernet ports. In this thesis, the UART is used to remotely access the board and one Ethernet port is connected to the WIA-FA devices.

Petalinux

A first necessary configuration step is to modify the standard board support package of the Zynq, which describes the board hardware design. Specifically, the CAN I/O port must be activated and then mapped to the J58 output. This allows to enable the can0 interface in Petalinux.

To customize the OS, first the Petalinux Installation Tools must be installed on a Ubuntu machine, then it is possible to build the Xilinx kernel present in the Github repository [30] by following the instructions in [31]. This step is crucial because it allows to select drivers and features that will be included in the Petalinux's kernel. In particular, the CAN drivers must be activated because they are not present by default.

The next step is to choose which software libraries to install in the operative system. Indeed, a list of most common utilities already cross-compiled for Petalinux is provided in the installation tool. Specifically, what is essential for the bridging operations are:

- SocketCAN can-utils package, a tool to manage the CAN traffic [32];
- Python compiler version 3.5;
- The SSH protocol library;
- TCPdump.

When the OS configuration in the Ubuntu machine is completed, an SD card is created, with two EXT4 partitions, BOOT and root. The former includes the BOOT file and the kernel; in the latter the whole Linux file-system is present with all the libraries and software previously selected. All the files created by the user and modifications are saved in the root partition and will not be erased after a system shutdown.

To remotely access the Zynq board from an external PC, the UART port is used that allows to access the OS command line. This serial port must be set with baud rate equal to 115200, Data as 8bits, no Parity and no Flow Control.

The setup includes two Zynq boards, one connected to the F3 and one to the F7, both remotely accessed through the same external PC through UART. Since it is often required to run the same command simultaneously on the two platforms, an open-source terminal emulator programmed in Java, called Terminator, is used to ease the process. [33]

6.2 Setup architecture

While the normal setup would include one CAN network connecting the F7 and all four F3s, in the bridged architecture considered in this thesis, the CAN network is split into two subnetwork, one with the F7 and the other with the F3s. The CAN traffic coming from the controller is redirected to a Zynq board, one for each subnetwork. The boards will collect the CAN packets from one interface and it will forward them to the other subnetwork through WIA-FA.

At the beginning, a point to point architecture is realized in which only F3 slave is present, as shown in Fig. 6.3.

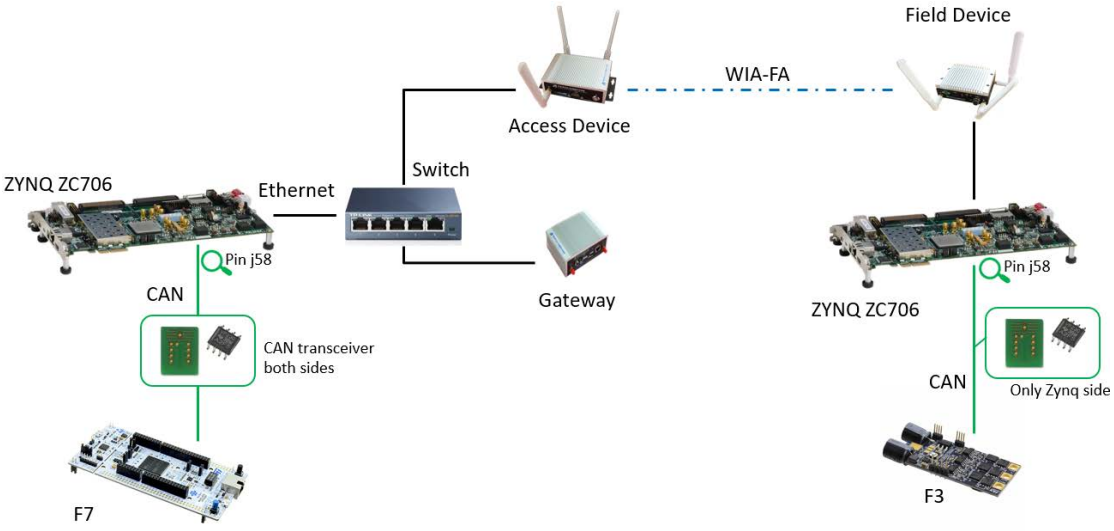


Figure 6.3: Point to point architecture.

On the left, there is the master side, the Zynq board is connected to F7 through CAN from the J58 pin and via Ethernet to the WIA-FA access device through the switch. On the right, the slave side, the Zynq board is linked to F3 through CAN from the J58 pin and to the WIA-FA field device via Ethernet. An external PC, not shown in Fig. 6.5, is connected to the two Zynq through UART.

After this preliminary assessment, an architecture more similar to the real one with one master and 4 F3s controllers is put into practice and it is represented in Fig. 6.4. By adding F3s it is necessary to modify the CANbus: three more male D-sub 9 must be connected to one of

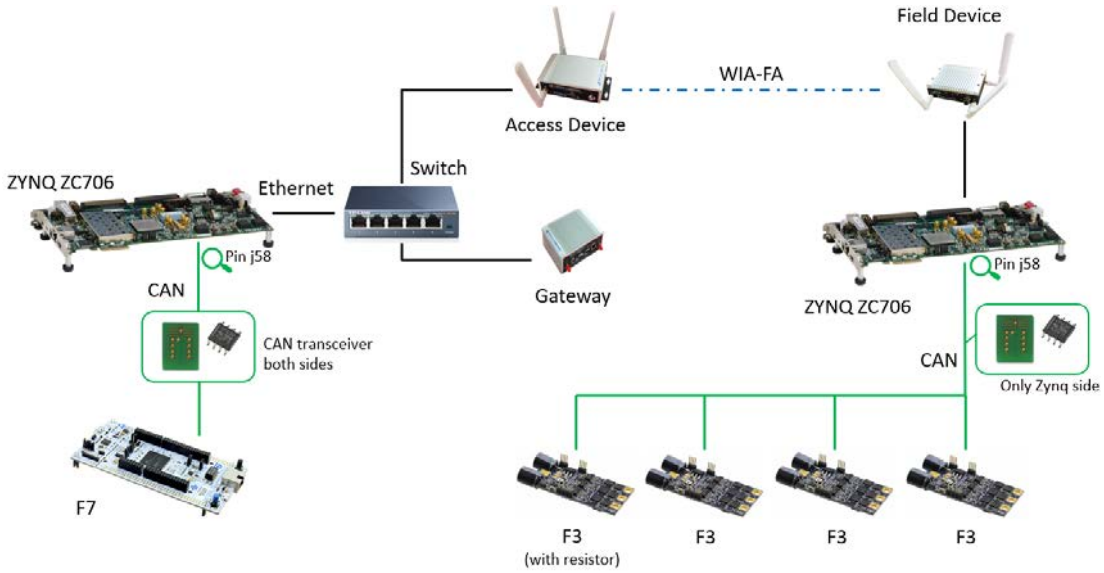


Figure 6.4: Point to multipoint architecture.

the end of the old bus by soldering new wires from pin 3 to pin 3 and the same for pins 4. A generic example for a bus with only two F3s is depicted in Fig. A.6 in Appendix A.

Moreover it must be remembered that only the first and the last controller must be terminated to avoid reflections. Hence the resistor named R2 in Fig. A.1 is removed from all the intermediate controllers in the bus, as shown at the bottom of Fig. 6.4.

Tha lab setup is presented in Fig. 6.5, where in the left there is the master side with the F7, access device, gateway and switch, instead, in the right, the slave side with the F3s and the field device. Both the CANbus present a yellow wire for CANH and a green one for CANL.

WIA-FA is set to operate with layer 3 mode for both architecture and implements GACK-based retransmission, enabled aggregation but no prioritization.

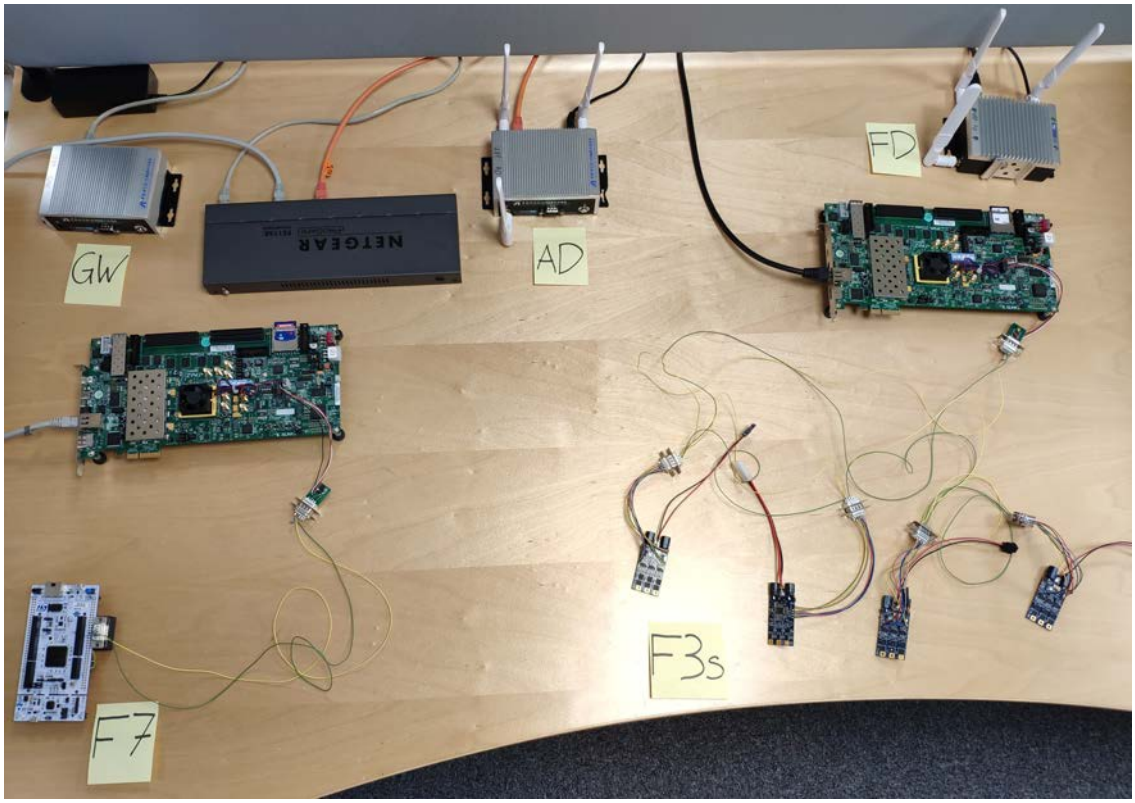


Figure 6.5: Point to multipoint lab setup.

6.3 Results

Different possible bridging strategies are considered for both architectures, their relative explanations and implementations as well as their performance evaluations and limitations are given.

6.3.1 Point to point architecture

A first step has been to evaluate the bridging in the point-to-point architecture with half-duplex communication only, meaning that only one of the two devices (F3 and F7) was active, while the other was silent. First, CAN packets only from F3 are forwarded to F7, as represented in Fig. 6.6a, then, the viceversa is put into practice and it is shown in in Fig. 6.6b.

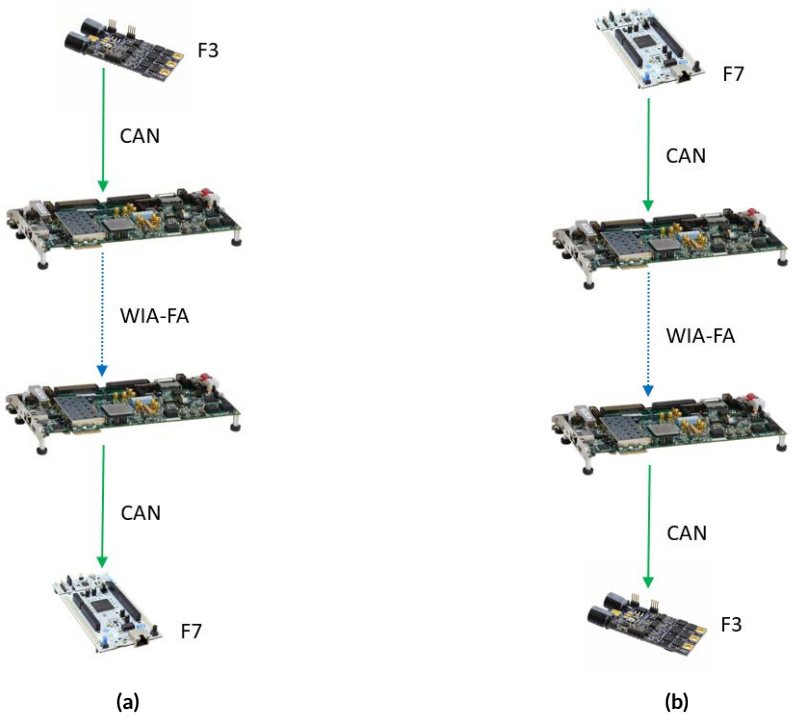


Figure 6.6: Half-duplex communication: (a) from F3 to F7; (b) from F7 to F3.

This preliminary step is useful to get an overview of the packets involved and their characteristics.

A Python script is run on the Zynq board connected to the active controller, it constantly listens to the CAN interface and as soon as a packet arrives it is immediately forwarded through WIA-FA to the other subnetwork. The other Zynq board is continuously listening the WIA-FA interface and whenever a packet arrives it is acquired and forwarded to the CAN interface.

Once the communication is established, the CANdump command from the can-utils tools is launched simultaneously on both platforms, thanks to the broadcast function of Terminator, in order to save all the CAN packets in a log file.

CANdump is executed for a specified amount of time, usually 180 seconds, and in the generated log file it is possible to check ID, DLC and payload of each CAN packet. Hence, the logs are analyzed with ad-hoc parser realized with Python, reporting the amount of packets for each ID.

Packets generated by F3 and sent to F7 are expressed in table 6.1. It is possible to see that the critical feedback packet from F3 has ID equal to 2 and the slow rate one has ID equal to 3. Conversely, packets generated by F7 and sent to F3 are expressed in table 6.2. It can be seen that critical packets from F7 have ID 1, 5, 9 and D. The debugging ones have IDs 28 and 29. It is worth to remind that the lower the ID, the higher the priority of the packet on the CANbus.

ID	2	3
PERIOD	2 ms	20 ms
TYPE	Feedback (Critical)	Feedback

Table 6.1: Packets generated by F3.

ID	1	5	9	D	28	29
PERIOD	2 ms	2 ms	2 ms	2 ms	2 ms	2 ms
TYPE	Command (Critical)	Command (Critical)	Command (Critical)	Command (Critical)	Debug	Debug

Table 6.2: Packets generated by F7.

After this first step, only needed to characterize the traffic, a proper bidirectional communication is performed by means of modified version of the previous Python scripts. The script

running on the Zynq on the F7 side collects a CAN packet which is forwarded via WIA-FA and waits for a reply from the other side; then the received packet is sent to the CAN interface. On the F3 side instead, the script collects a packet from WIA-FA which is sent to the CAN interface; a CAN reply is then prepared and transmitted to the the other side. The sequence of steps is summarized in Fig. 6.7.

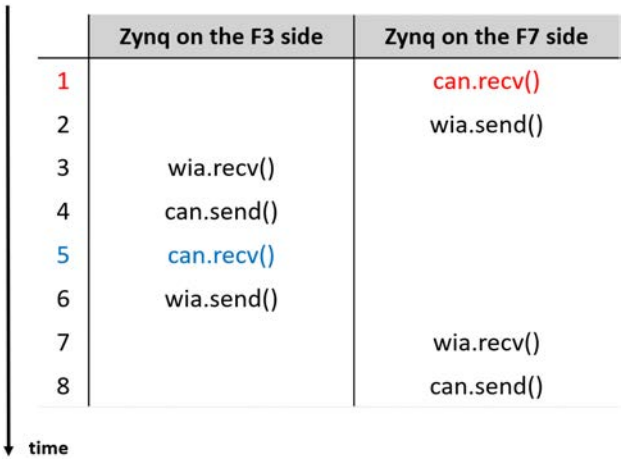


Figure 6.7: Pseudocode for duplex transmission.

After a couple of attempts, it has been observed that it was impossible to established a proper communication when the WIA-FA devices were 0.5-1 meters distant. After separating the devices with a distance of 2 meters, as recommended by the WIA-FA standard [26], the communication has not showed any problem.

To assess the quality of the packet loss rate is computed according to the following formula:

$$\frac{\sum_{x \in ID} Gen_x - Rec_x}{\sum_{x \in ID} Gen_x}, \tag{6.1}$$

in which Gen_x indicates the number of generated packet for ID x during the dump and Rec_x indicates the number of received packets for the same ID. The formula can be computed over all the possible IDs or just for the critical ones, depending on the quantity under observation.

Results for a 180 seconds dump are given in Fig. 6.8. This graph compares the dump from

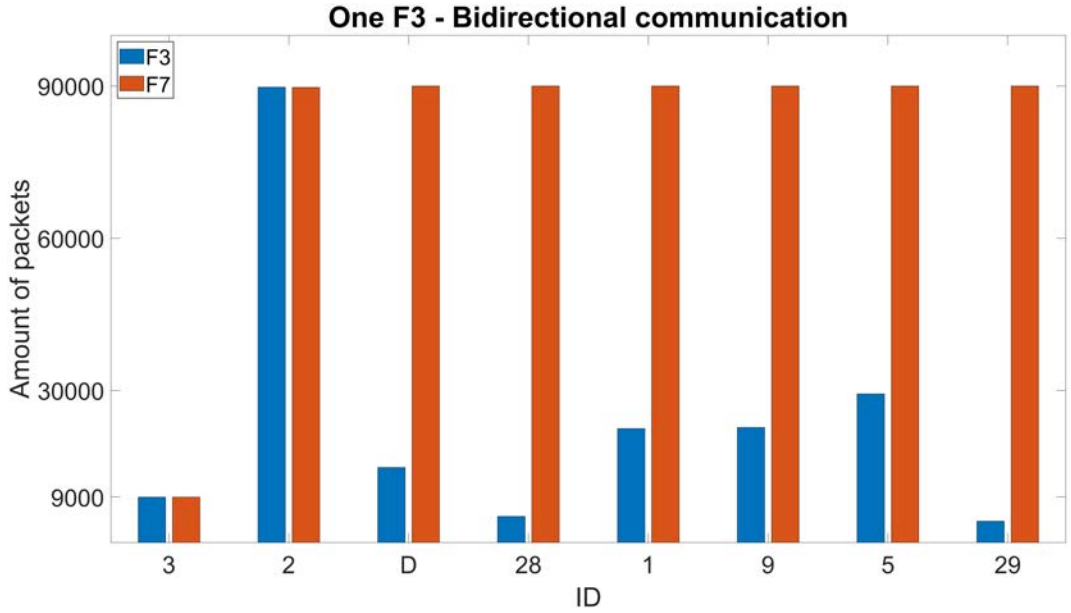


Figure 6.8: Dumped packets with one F3 slave, full duplex communication.

the platform connected with F3, blue bars, and the dump from the Zynq connected with F7, red bars.

Since IDs 2 and 3 are produced by the F3, the blue bars indicate the amount of generated packets and the red ones the amount of delivered packets to F7. Instead, the remaining IDs are produced by F7, so the red bars are the total generated packets for each ID and the blue ones the delivered to F3. The gap between the two bars for each ID indicates the losses. This logic will be used also in the following bar plots in the remainder of the thesis.

It is visible that a huge quantity of packets have been lost during the transmission, in fact, the overall loss rate is 69%; the exact numbers for each message ID are shown in Table 6.3.

ID	3	2	D	28	1	9	5	29
F3	8977	89772	14809	5193	22488	22701	29323	4236
F7	8974	89738	90000	90000	90000	90001	90001	90001
Losses	3	34	75191	84807	67512	67300	60678	85765
%	0.03%	0.03%	83.50%	94.20%	75%	74.70%	67.40%	95.30%

Table 6.3: Performance with one F3 slave, bidirectional communication.

After analyzing these results, it emerges that the packets generated by F3 are almost completely delivered, while this is not the case for the generated by F7. To understand the reasons of this behavior, it is helpful to look at the arrival times of the packets, depicted in Fig. 6.9.

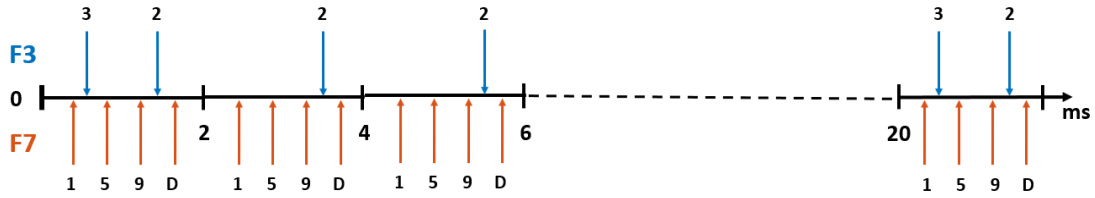


Figure 6.9: Arrival times for point to point architecture.

Considering the period of the critical and non-critical packets, it can be said that the time is divided into slots of 2 ms. From the F7 side, in each timeslot there are the four command packets (from now on the debugging packets are omitted for simplicity). On the F3 side, the critical packet is present in each time slot and the non-critical is repeated every 20 ms.

This analysis allows to conclude that the code explained in 6.7 is not properly designed for the architecture. Indeed, in step 5, the script is waiting for a CAN packet from F3, which arrives every 2 ms (with some exceptions due to the non-critical packets) and in the meanwhile packets keep coming from F7. But, since the algorithm is hanging on step 5 until an arrival, F7 packets are wasted.

Hence, the two sides must be synchronized and a first solution is provided by merging several CAN packets into one WIA-FA frame. That is, the Zynq board connected to F7 collects exactly four packets, merges them in a unique message which is forwarded to the other subnetwork. When this merged frame is received by the board connected to F3, it is split into the original CAN packets which are subsequently sent to the CAN interface.

This is feasible because both the CAN packet and the WIA-FA MAC header lengths are well known: the first is 16 Bytes and the second is 14 Bytes. Therefore the merged frame, including 4 CAN packets and one WIA-FA header, contains 78 Bytes.

This strategy has been adopted for a new experiment session of 180 seconds duration. Results are provided in Fig. 6.10. As can be seen, now all the critical bars are almost aligned and

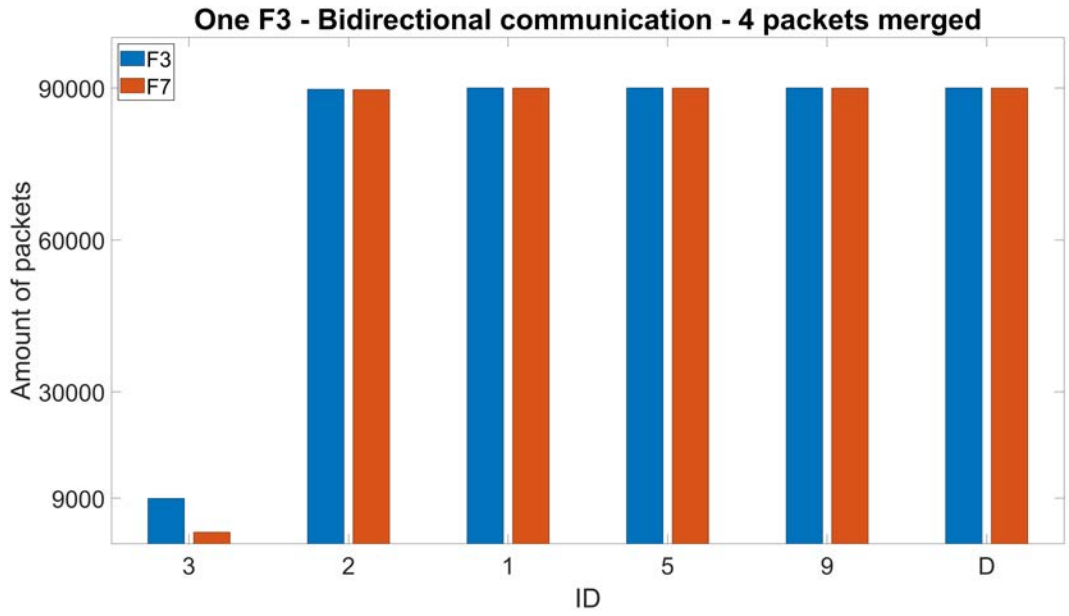


Figure 6.10: Dumped packets with one F3 slave, full duplex communication, 4 packets merged at master.

the majority of the packets lost the non-critical ones. A closer look to the results is given in Table 6.4. The total loss rate is 1.5%.

ID	3	2	1	5	9	D
F3	8976	89757	90001	90001	90001	90001
F7	2305	89695	90022	90022	90022	90022
Losses	6671	62	21	21	21	21
%	74.32	0.07	0.02	0.02	0.02	0.02

Table 6.4: Performance with one F3 slave, bidirectional communication, four merged packets at master.

6.3.2 Point to multipoint architecture

The next step in the experiments has been to evaluate a point-to-multipoint bridging with one F7 master and four F3 slaves, as shown in Fig. 6.4. All the new F3s behave like the previous one and the IDs of their generated packets are given in Table 6.5, 6.6 and 6.7.

It is important to figure out the generation times for the new configuration in order to provide an effective solution. The behavior is shown in Fig. 6.11: the F7 generation is unchanged, whereas F3 constantly presents four critical packets in a timeslot and an undefined number

(up to 4) of non-critical packets. The uncertainty related to those packets makes difficult to optimally synchronize the two sides because it is challenging to predict when one of them will occur.

ID	A	B
PERIOD	2 ms	20 ms
TYPE	Feedback (Critical)	Feedback

Table 6.5: Packets generated by the second F3.

ID	6	7
PERIOD	2 ms	20 ms
TYPE	Feedback (Critical)	Feedback

Table 6.6: Packets generated by the third F3.

ID	E	F
PERIOD	2 ms	20 ms
TYPE	Feedback (Critical)	Feedback

Table 6.7: Packets generated by the fourth F3.

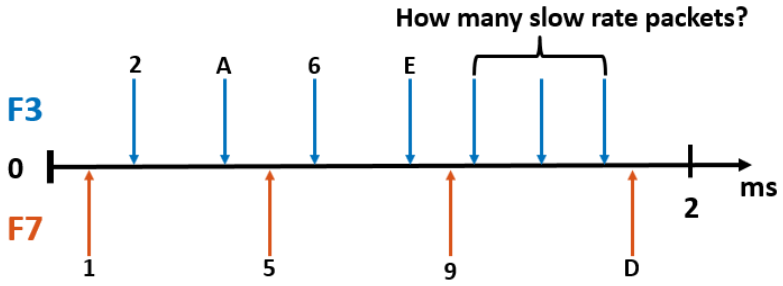


Figure 6.11: Packet arrival times for point to multipoint architecture.

Three different scheduling strategies are proposed and investigated in the following:

- 1) critical IDs only,
- 2) merging solution;

- 3) adaptive solution.

Critical IDs only

In this strategy all the non-critical packets are filtered out, meaning that IDs 3, B, 7 and E are excluded from the communication. The filtering action is applied at kernel level, so this does not impact the communication. It must be specified that not forwarding non-critical packets may impact the system and this must discussed with developers of the controller software. It is out of the scope to investigate the the effects of this strategy.

Now both sides generate four critical packets in a timeslot creating a balanced situation. The algorithm is similar to the previous, but now four packets collected from the CAN interface and merged in one WIA-FA frame, on both sides.

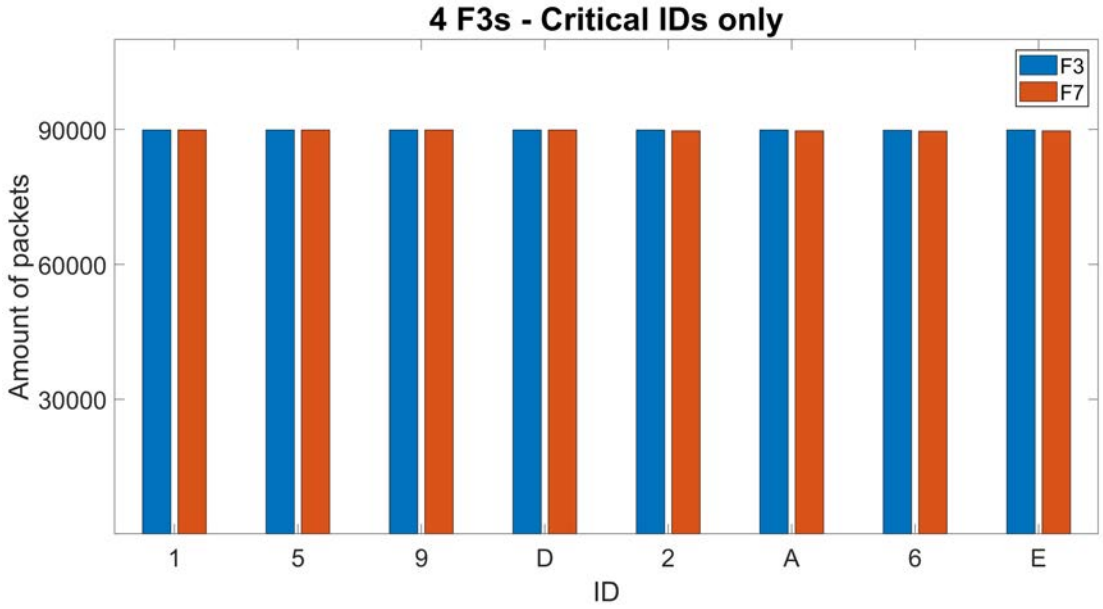


Figure 6.12: Dumped packets with 4 F3 slaves, critical IDs only.

The results for a 180 seconds dump are shown in Fig. 6.12 and Table 6.8. The WIA-FA time slot is set to 128 μs. The percentage of lost critical packets is 0.12%, while the overall loss rate drops to 4.88% if it is taken into account that the loss for non-critical packets is 100% due to the filtering. The residual lost packets are likely due to the delays in the forwarding process

ID	1	5	9	D	2	A	6	E
F3	89888	89889	89889	89888	89892	89896	89848	89904
F7	89893	89893	89892	89918	89696	89673	89627	89680
Losses	5	4	3	30	196	223	221	224
%	0.005	0.004	0.004	0.004	0.24	0.24	0.24	0.24

Table 6.8: Performance with 4 F3 slaves, critical IDs only.

between the CAN and WIA-FA interfaces.

To further optimize the performance, the experiment is repeated for WIA-FA timeslot lengths of 64, 128 and 256 μs. Since this application is a preliminary work and considering that a 180 seconds communication is quite long, a trade-off between level of confidence and total amount of time to perform the whole experiment is needed. In fact, tests are repeated only 12 times for each timeslot. This assumption is adopted also for the next assessments.

The statistics are given in Table 6.9. It emerges that a timeslot length of 128 μs provides the best results.

Timeslot	256 μs	128 μs	64 μs
Average	$4.98 \cdot 10^{-2}$	$2.2 \cdot 10^{-3}$	$2.8 \cdot 10^{-3}$
Variance	$3.19 \cdot 10^{-6}$	$2.3 \cdot 10^{-6}$	$5.4 \cdot 10^{-6}$
STD	$0.18 \cdot 10^{-2}$	$1.5 \cdot 10^{-3}$	$2.3 \cdot 10^{-3}$

Table 6.9: Loss rate with 4 F3 slaves, critical IDs only, for different WIA-FA timeslot lengths.

Merging solution

This solution is similar to the merging adopted for the point to point architecture. Four packets from F7 are merged, whereas a fixed amount of packets is merged from the F3 side for all the duration of the communication. This number can be either four or five.

The solution with 4 merged packets at the slave side and a WIA-FA timeslot of 128 μs is first investigated. Results are reported in Fig. 6.13. At a first glance, it is clear that packets generated by F7 are almost completely delivered, whereas many losses occurred for those generated by F3s.

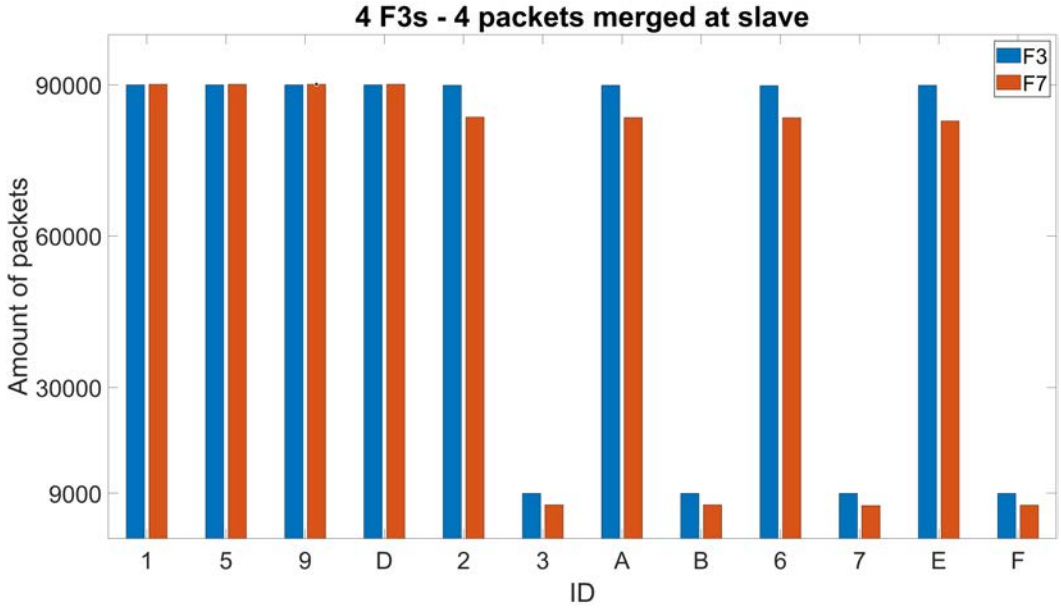


Figure 6.13: Dumped packets with 4 F3 slaves, four packets merged at slave.

Full results are shown in Table 6.10, highlighting how F7 packets are almost totally delivered and the amount of all critical packets lost is 3.7%. Although the non-critical loss is 26%, the overall loss is 4.8%, because the slow rate packets are significantly less than the criticals ones. The assessment is repeated merging five packets from F3s. This method gives advantage to

ID	1	5	9	D	2	3	A	B	6	7	E	F
F3	90001	90002	90001	90001	89929	8993	89906	8991	89858	8986	89912	8991
F7	90125	90124	90124	90125	83586	6680	83520	6710	83480	6557	82833	6638
Losses	124	122	123	124	6343	2313	6386	2281	6378	2429	7079	2353
%	0.13	0.13	0.13	0.13	7	25.7	7	25.3	7	27	7.8	26

Table 6.10: Performance with 4 F3 slaves, 4 packets merged at slave.

the F3s since their critical and non-critical packets bars are almost aligned. Several losses occurred instead for the packets generated by the F7.

A further look to the results can be given in Table 6.11. The 6.12% of the critical packets have been lost, whereas the non-critical ones count a loss 0.12%. The first result highly impacts on the performance of the communication becuase the overall losses are now 5.7%, sightly worse than the other merging solution. These two experiments highlight how it is not possible to find a fixed amount of packets to merge that achieves optimal performance. Therefore, if it

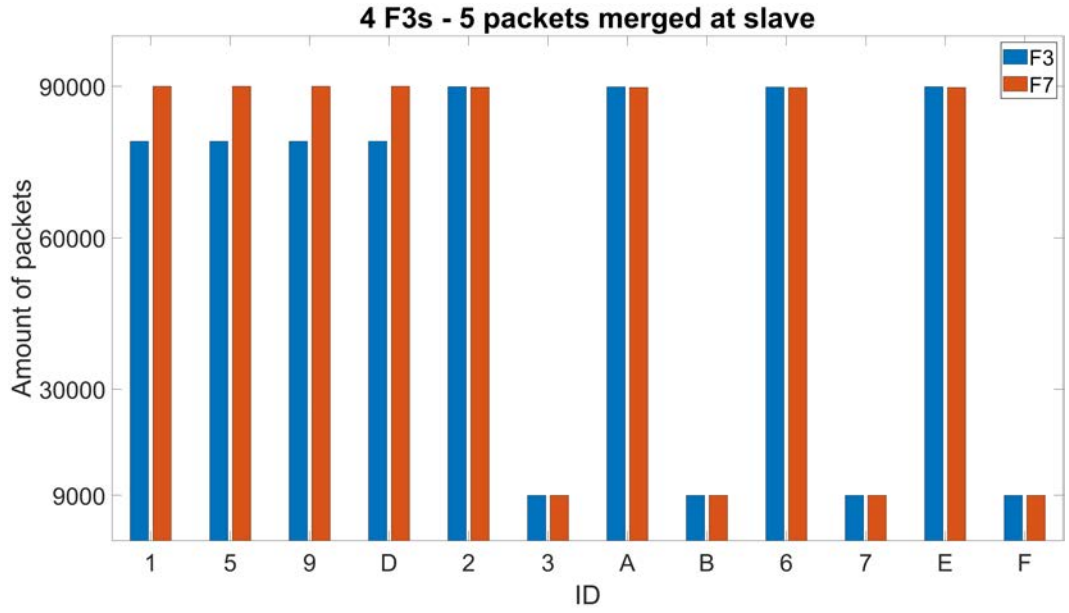


Figure 6.14: Dumped packets with 4 F3 slaves, five packets merged at slave.

ID	1	5	9	D	2	3	A	B	6	7	E	F
F3	79108	79109	79108	79108	89924	8992	89900	8990	89854	8985	89907	8990
F7	90004	90003	90003	90004	89807	8981	89784	8978	89736	8974	89791	8979
Loss	10896	10894	10895	10896	117	11	116	12	118	11	116	11
%	12.1	12.1	12.1	12.1	0.13	0.13	0.12	0.12	0.13	0.13	0.12	0.13

Table 6.11: Performance with 4 F3 slaves, 5 packets merged at slave.

is not desirable to filter out non-critical packets, a third solution must be developed.

Adaptive solution

The idea behind this solution is that the CANbus and WIA-FA packets are handled in two different threads. In this way, both data flows are handled independently. The algorithm, which has been realized exploiting the multithreading Python library, is intuitively described in Fig. 6.15.

Two buffers, A and B, are defined at the beginning of the script as global variables, in which CAN packets and WIA-FA packets are respectively stored; they can be accessed by both flows. In the CAN thread, incoming packets from the interface are collected for 2 ms and aggregated together. At the beginning of this merged packet, a counter byte is appended to indicate how many packets have been merged to simplify the future disaggregation.

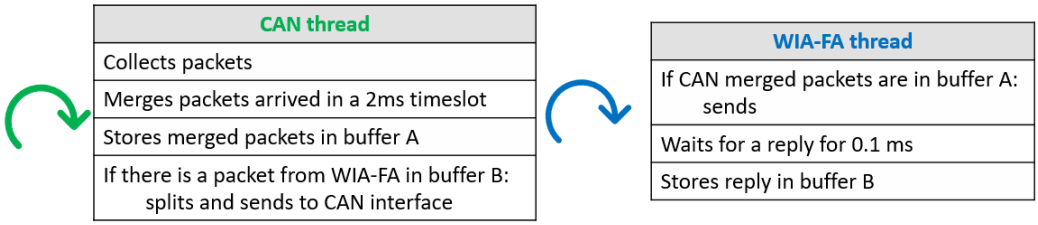


Figure 6.15: Adaptive solution scheme.

The aggregated packet is put into buffer A, then the CAN thread checks if buffer B is empty. If not, this means that at least one packet has arrived from WIA-FA, so it is split into the original messages, that are sent to the CAN interface. At this stage, the loop restarts.

The WIA-FA thread checks the presence of a merged CAN packet in buffer A. If there is at least one, then it transmits it to the other board, otherwise it just skips. At this step, WIA-FA waits for 0.1 ms for a reply and if any packet arrives then it is inserted in buffer B, stripped of the MAC header. If WIA-FA has been waiting for 0.1 ms without receiving anything, then the loop restarts.

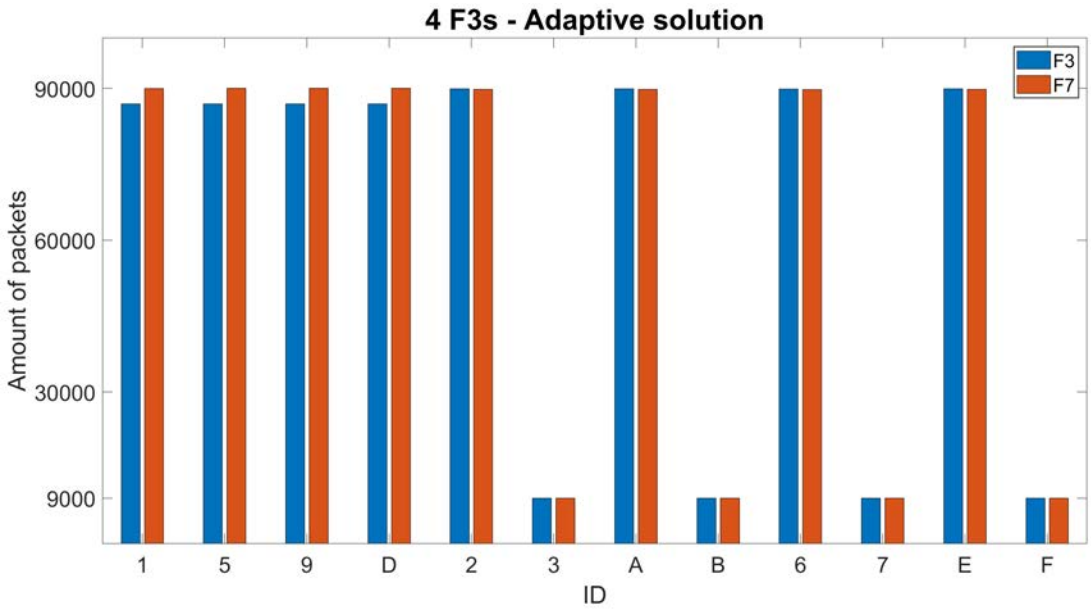


Figure 6.16: Dumped packets with 4 F3 slaves, adaptive solution.

The results for a dump of 180 seconds and WIA-FA timeslot equal to 128 μs are presented in Fig. 6.16. A significant improvement of the performance from the merging solution can be

clearly observed. The exact results are given in Table 6.12.

	1	5	9	D	2	3	A	B	6	7	E	F
F3	86914	86947	86948	86947	89929	8993	89910	8991	89859	8986	89915	8990
F7	89967	90001	90000	90001	89809	8980	89789	8979	89739	8974	89795	8980
Losses	3053	3054	3052	3054	120	13	121	12	120	12	120	10
%	3.39	3.39	3.39	3.39	0.13	0.14	0.13	0.13	0.13	0.13	0.13	0.11

Table 6.12: Performance with 4 F3 slaves, adaptive solution.

Indeed, only 1.76% of the critical packets and 0.13% of the non-critical ones are lost. The overall loss is 1.68%, yielding a significant improvement over the previous solution.

The experiment is repeated 12 times for WIA-FA timeslot lengths 64, 128 and 256 μs to assess which is the best suited for the solution. The statistics are presented in Table 6.13.

Timeslot	256 μs	128 μs	64 μs
Average	1.78 x 10^{-2}	1.65 x 10^{-2}	1.11 x 10^{-2}
Variance	9.17 x 10^{-5}	4.92 x 10^{-5}	3.69 x 10^{-5}
STD	0.9 x 10^{-2}	0.7 x 10^{-2}	0.6 x 10^{-2}

Table 6.13: Loss rate with 4 F3 slaves, adaptive solution, for different WIA-FA timeslot lengths.

In this case, a timeslot length of 64 μs shows lower average errors and lower STD than the other options.

6.3.3 Experiment by emulated traffic

A final experiment has been conducted in which the traffic generated by the F3 and F7 slaves is replicated on the Zynq boards and directly sent over the WIA-FA networks. In this way, a full control on the process is obtained and high level of synchronization is achieved.

It is important to highlight that in this final experiment there was no communication over CAN. The reason to perform this experiment is to test the best achievable performance, once the problem of synchronizing the WIA-FA and CAN interfaces is removed.

One board mimics the F7, it generates the four critical ID packets, each of them every 2 ms. The other board copies the behavior of four F3s, producing the critical packets, each of them every 2 ms, and four non-criticals ones, each of them every 20 ms.

A packet is made of 30 bytes: 14 as WIA-FA header and 16 as payload, the same length of a CAN packet. In the payload, the first byte is dedicated for the ID and the remaining 15 is a dummy fixed string.

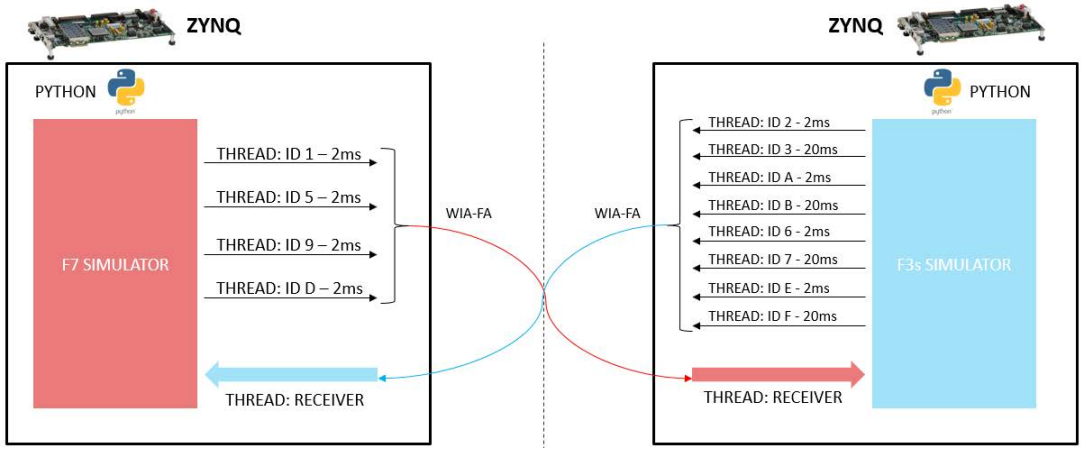


Figure 6.17: Representation of the emulated traffic flows.

Every ID is considered as an independent flow, since they are generated in different threads. In addition, one more thread is used to collect the incoming packets from the other side. This is put into practice exploiting the Python multithreading library and graphically described in Fig 6.17.

Results for this experiment performed in 180 seconds with WIA-FA timeslot of 128 μs are given in Fig. 6.18; the detailed communication results are given in Table 6.14.

ID	1	5	9	D	2	3	A	B	6	7	E	F
F3	90448	90421	90440	90464	90294	9143	90289	9143	90321	9143	90275	9143
F7	90452	90426	90446	90468	90294	9143	90287	9143	90318	9143	90274	9143
Loss	4	5	6	4	0	0	2	0	3	0	1	0
%	0.004	0.005	0.006	0.004	0	0	0.002	0	0.003	0	0.001	0

Table 6.14: Performance with 4 F3 slaves, emulated traffic.

It possible to see that all the non-critical packets are successfully delivered and at most 6 critical packets are lost. The overall loss is 0.003%, yielding a very significant improvement with respect to the previous tests.

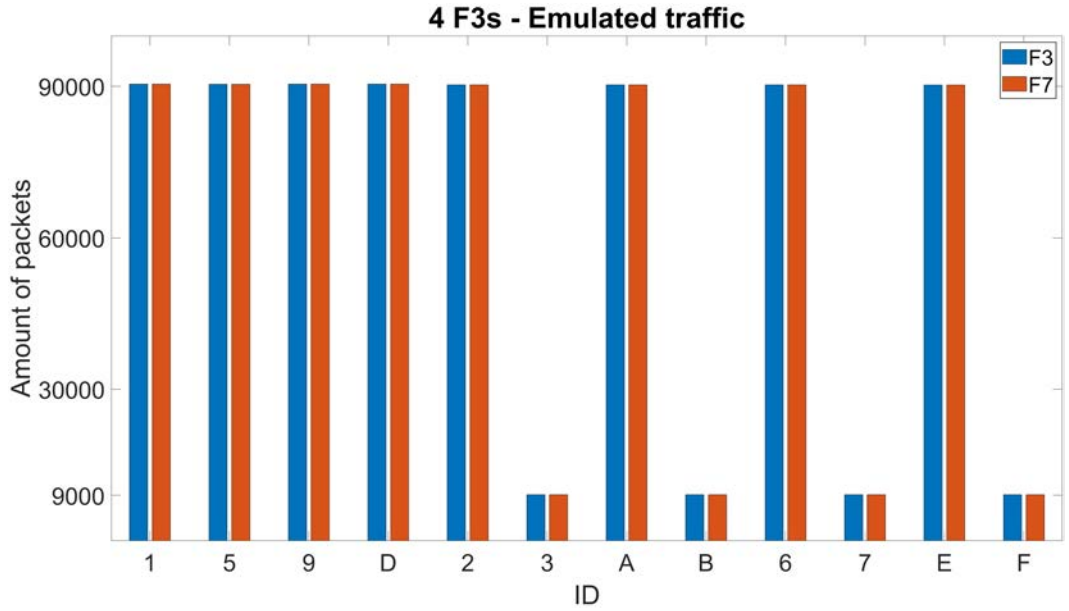


Figure 6.18: Dumped packets with 4 F3 slaves, emulated traffic.

One could notice that the amount of generated packets for each ID is slightly higher than expected, this is due to the fact that time is not perfectly managed. A more low level implementation can improve the accuracy of the experiment.

The experiment is repeated 12 times for WIA-FA timeslot lengths of 128 and 256 μs, with the results reported in Table 6.15.

Timeslot	256 μs	128 μs
Average	11.2 x 10^{-4}	2.5 x 10^{-5}
Variance	4.2 x 10^{-7}	5.3 x 10^{-10}
STD	6.5 x 10^{-4}	7.2 x 10^{-5}

Table 6.15: Loss rate with 4 F3 slaves, emulated traffic, for different WIA-FA timeslot lengths.

Timeslot length of 128 μs gives the best performance, with average and STD loss rate in the order of 10^{-5}.

6.4 Conclusion

For this preliminary experimental study, it can be concluded that a CAN network, employed for the control of a mobile robot equipped with one master and 4 slaves, can be extended by means of WIA-FA.

This hybrid network (CAN + WIA-FA) is able to successfully deliver the most of the critical packets, which are periodically generated every 2 ms, with a loss rate of 0.12%.

If the non-critical packets, which are periodically generated every 20 ms, are included in the communication, a degradation of the performance is observed due to an imperfect synchronization between the arrival rates of the CANbus and WIA-FA packets. By using an adaptive solution based on multithreading, the rate of the lost packets has been reduced to 1.68%.

In the final experiment, in which the CANbus network is removed and the same generated packets are directly sent through WIA-FA, a packet loss rate of 0.003% can be achieved. This allows to conclude that the main problem of the hybrid network is not related to the performance of the wireless part (WIA-FA). In fact, the lack of a synchronization and the delays between the processes that generate packets and transmit them through CAN and the processes that receive the packets and send them through WIA-FA, are the major limiting factors.

Flectere si nequeo superos, Acheronta movebo.

Virgil, Aeneid, VII, 312

7 Future technologies

As discussed in Chapter 2 and demonstrated in the experimental evaluation, the currently available wireless solutions do not fully satisfy the requirements for a deterministic and real-time communication as demanded by smart manufacturing scenarios. In cases where such performance are required for mobile devices, it is hence necessary to use proprietary solutions (such as the WIA-FA considered in this thesis) which, however, lack the interoperability provided by major wireless standards.

The major wireless standardization bodies view the gap present in this sector as a big opportunity. Indeed, in the next few years, many new wireless technologies are expected to provide solid solutions to bridge this gap by offering higher performance.

It is the scope of this section to investigate the most promising technologies, describing their main features and highlighting why they are appealing for industrial cases and also what are their possible flaws.

The technologies discussed are 5G, the IEEE802.11ax standard and the possibility to integrate TSN functionalities in wireless networks.

7.1 5G

Mobile industries have invested a lot in research efforts during the last years toward the designing of the Fifth Generation mobile architecture. Such a technology, finally available from 2019, does not only aim to enhance the mobile user experience, but targets to provide innovative capabilities and solutions for several use cases and scenarios. Network slicing is one of proposed innovation to push forward the frontier of mobile communications.

5G network slicing permits Mobile Network Operators (MNO) to share the owned physical infrastructures among simultaneous deployment of multiple independent logical networks, managed with respect to their specific service requirements. For example, different services like automotive, tactile internet or massive IoT can be put in practice by exploiting different network slice instances, that is a set of virtual network functions that run on the same infrastructure with customized policies.

Ultra-reliable and low-latency communications is the slice that will address the strict requirements dictated by most industrial uses cases. It aims to provide latency of 1 ms and reliability of 99.9999% for most demanding scenarios. The main use cases served by this technology and their requirements are summarized in Table 7.1.

Scenario	End-to-end latency	Reliability
Discrete automation – motion control	1 ms	99.9999%
Electricity distribution – high voltage	5 ms	99.9999%
Remote control	5 ms	99.999%
Discrete automation	10 ms	99.99%
Intelligent transport systems – infrastructure backhaul	10 ms	99.9999%
Process automation – remote control	50 ms	99.9999%
Process automation – monitoring	50 ms	99.9%
Electricity distribution – medium voltage	25 ms	99.9%

Table 7.1: Low latency and high reliability for different use cases. [34]

Next Generation Mobile Networks (NGMN) designed the concept of network slicing as ideally composed by Service Instance Layer (SIL), Network Service Instance Layer (NSIL) and resource layer. The SIL involves the business/end user services provided by operators or by the party which leases the services from the operators, the NSIL is a set of functions to run the instances and, finally, the resource layer consists of the resources such as computing

power, communication bandwidth, memory and storage.

The target of slicing is to realize End to End (E2E) network slices from the mobile edge, through the mobile transport (Fronthaul (FH)/Backhaul (BH)) and up until Core Network (CN).

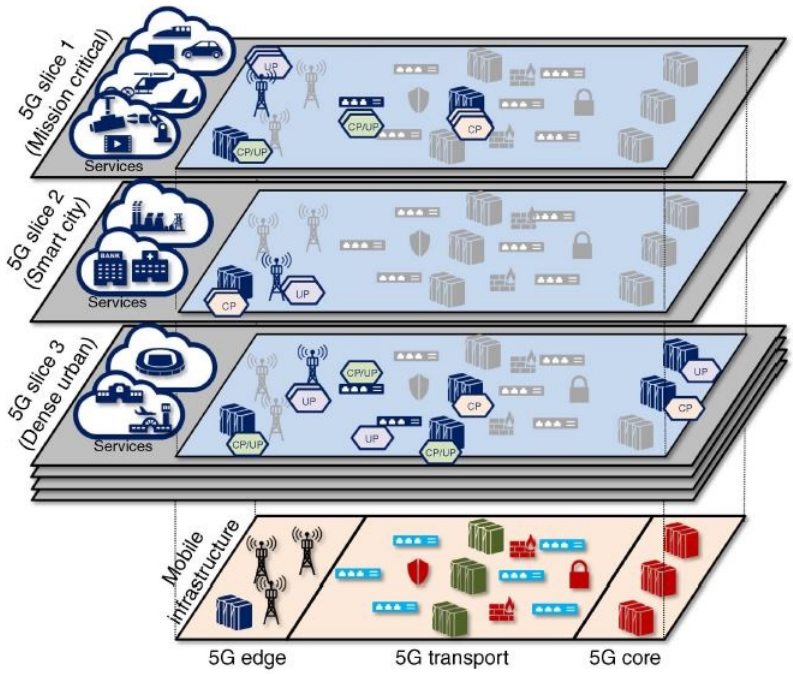


Figure 7.1: Network slicing in 5G networks. [35]

The concept is depicted in Fig. 7.1, in which the resources at the resource layers are dimensioned to create many subnetwork instances and network slice instances.

7.1.1 Requirements and enabling technologies

Future 5G networks will be built on novel technologies with respect to the previous generation network architectures. One of the main innovations is provided by the introduction of Software-Defined Networks (SDN) and Network Functions Virtualization (NFV). The former focuses on separating the control and data planes for a centralized view of the network, instead the latter on optimization of the network services.

For the sake of clarity, the data plane is actually the resource layer where the mobile edge,

mobile transport, and core are present. The control plane comprehends the Management and Orchestration (MANO) which executes control and management functions on the actual hardware components.

Modularization and function decomposition

The first step towards modularizing the architecture is put into practice by means of Network Functions (NF)s, entities that furnish specific network capabilities in order to realize and support the requested services. Generally they are software instances acting on infrastructure resources, but they can also be physical instances, a combination of them or virtualized, that is decoupled from the hardware they run on.

The general network functions are proposed to be split into basic modules, both for the Control Plane (CP) and User Plane (UP), allowing the definition of different logical architectures by means of the interconnection of different subsets of NFs for CP and UP.

To realize the highest level of decomposition possible, it is necessary to distinguish between NFs of the Access Network (AN) and CN, in order to achieve what is called a convergent network. This means the coexistence of different kind of information to transmit, such as voice, video and data, within a single network implies that AN/CN split is mandatory to to support network slicing.

Virtualization, orchestration and isolation

Virtualization is the main process in the network slicing architecture, because it allows to share effectively the resources among the various slices and it operates the abstraction of resources. Abstraction means the representation of the underlying resources in order to recreate a virtual scenario with same peculiarities.

The resources to be virtualized can be physical or already virtualized, generating a recursive structure in the system counting different abstraction levels. Exactly like server virtualization makes Virtual Machines (VM)s free from the physical hardware, network virtualization allows to generate multiple isolated virtual networks, completely independent from the physical networks over which they can run.

In a scenario, where the entities involved are so heterogeneous, the so called orchestrator, is needed to coordinate and manage the different services related to all the assigned requirements.

Quoting the Open Networking Foundation (ONF) definition, orchestration is: "*the continuing process of selecting resources to fulfill client service demands in an optimal manner*" [36]. This means that a policy to handle the orchestrator behavior is required and it is expected to satisfy the service level agreements with clients requirements. This policy also has to consider that the available resources, the demands and optimization criteria may change in time.

An effective isolation is required to let parallel slices run on a common underlying substrate. The isolation must specific service requirements of each slice, expressed generally as Key Performance Indicators (KPI), and security/privacy, that is attacks or issues in a slice must not affect other slices.

Software-defined network and network functions virtualization

In this section, the technologies that allow to get the requested level of virtualization are described.

In SDN, the admin or an engineer is able to handle the data traffic remotely exploiting a centralized control without acting directly on particular switches of the network. In this context, the SDN controller tunes the switches in order to deliver the Network Services (NS) wherever they are needed; this is a step away from the classical architecture, where devices take their traffic decisions based on routing tables.

SDN is the key technology to realize abstraction of the resources. Its main components are resources and controllers. A resource is defined as everything useful to provide services as answers to client requests, that in this case are infrastructure resources and NFs. A controller, instead, is the centralized entity, implemented at the control plane, that enables the virtualization of resources and orchestrates the process of assigning these virtualized resources.

An SDN architecture is not enough to enable slicing, because it lacks the capabilities to manage the network slices life cycles an the related resources. Virtual Network Functions (VNF),

move NFs out of specific hardware devices into software running on generic hardware; some classic examples include firewalls, domain name system and caching.

An architecture employing NFV is suitable to administrate infrastructure resources and to orchestrate the allocation of the necessary resources to get VNFs and NSs. The main problem in this domain is to ensure a proper coordination between SDN and NFV. ETSI recently presented a way to join a SDN within the NFV architecture and it is described in [37].

7.1.2 5G Architecture

3GPP will release 5G specifications into multiple successive releases, starting with, the Release-15 on August 2018 addressing related to commercial needs and Release-16, planned for March 2020 to address other use cases and related requirements. A representation of 5G architecture is shown in Fig. 7.2 and its functions are discussed in the following.

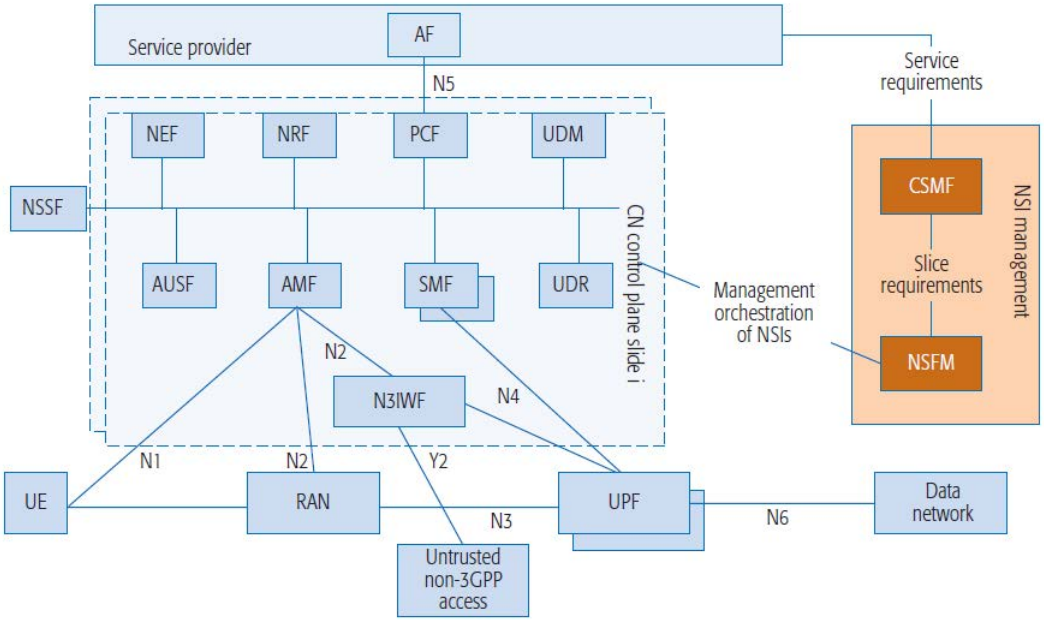


Figure 7.2: 5G System architecture. [38]

The control plane functions of the core network of a generic slice are:

- Core access and mobility management function (AMF), necessary to manage mobility, authenticate and give authorization, for what concerns security functions and context selection;

- Session management function (SMF), handling session management, allocation of IP addresses, selection of UP functions and QoS;
- Policy Control Function (PCF), providing an unified policy framework to govern network behavior and plane functions;
- Network Exposure Function (NEF), ensuring that the exposition of the services provided by NFs, that means guarantee the translation of the information received from the AF to the one sent to the NFs, and vice versa;
- Network Repository Function (NRF) is a service discovery function exploited by NFs;
- Unified data management (UDM), involving the repository where the authentication credentials are stored and the subscription management;
- Authentication server function (AUSF), supporting the authentication server;
- Non-3GPP InterWorking Function (N3IWF), necessary to support non-3GPP access networks;
- Unified Data Repository (UDR), storing subscription and policy data;

The architecture also contains the following functions:

- User plane function (UPF) is an anchor point for mobility, it is mandatory for packet routing and forwarding and defining QoS for the UP;
- Network slice selection Function (NSSF), necessary to tie a UE to a specific slice;
- Application Function (AF), influencing traffic routing, access to the NEF and interaction with PCF;

The management functions are called Communication Service Management Function (CSMF) and Network Slice Management Function (NSMF). The first translates the service requirements for communication to network slice requirements: capacity, throughput, delay, number of users, etc. The second takes care of the life cycle of a slice. 5G allows UEs to access multiple slices at the same time, but one only AMF will be involved for all slices; 8 slices in parallel is the limit.

7.1.3 3GPP standardized slices

The slices that have been standardized by 3GPP are:

- Enhanced mobile broadband (eMBB): it actually is an evolution of today 4G network and it is related to operation done by humans with the target to enhance access to multimedia and give services with improved performance towards an increasing Quality of Experience. It also has to support restricted high density user area, with huge traffic and poor mobility of users, and wide area coverage where user mobility is medium or high.
- Ultra-reliable and low-latency communications: it is necessary where the requirements for latency and reliability are very strict. It will play a fundamental role in the industry 4.0 use cases, such as remote medical surgery, automation industries and smart grids. URLLC aims to provide 1 ms of latency, support mobility for 500 km/h and high reliability.
- Massive machine-type communications (mMTC): it is the realization an efficient Internet of Things, a huge number of connected devices transmitting a low volume of data traffic, usually non sensitive of the delay which can be about seconds or hours. In general, these devices must be low-cost with a long lifetime. Some examples are logistics applications, smart metering and agriculture, where a lot of sensors are spread around to collect data.

The 8 most important KPIs defined in 5G are the following:

- Peak data rate: ideal maximum data rate achievable per user in bits per second. The minimum requirement is 20 Gbps in downlink and 10 Gbps in uplink;
- User experienced data rate, achievable data rate available on whole coverage area perceived by a mobile user in bits per second. It is set to 100 Mbps in downlink and 50 Mbps in uplink;
- Average spectral efficiency, it is the average data throughput per unit of spectrum resource and per cell in bps/Hz/cell. Here, the minimum requirement depends on the environment:
  - Indoor Hotspot: 9 bps/Hz/cell in downlink, 6.75 bps/Hz/cell in uplink;
  - Dense Urban: 7.8 bps/Hz/cell in downlink, 5.4 bps/Hz/cell in uplink;
- Rural: 3.3 bps/Hz/cell in downlink, 1.6 bps/Hz/cell in uplink.

- Area traffic capacity, only for indoor, it is the total traffic throughput per area in Mbps/m^2, aiming to supply 10 Mbps/m^2 for downlink;

- User plane latency, time from when the source sends a packet to the moment in which the destination receives it; it is set to 4 ms for eMBB and 1 ms for URLLC;

- Connection density, it is the total number of connected UEs per unit area, it targets 1 billion devices per km^2 for mMTC;

- Energy efficiency, it is the ratio between users bits transmitted/received and energy consumption of the RAN/device in bits/Joule.

- Mobility, it is the maximum speed at which a defined QoS are guaranteed while UEs is moving between base stations. The target in non-urban environments is, the normalized traffic channel link data rate is 500 km/h.

The following web-spider diagrams in 7.3 are useful to represent the situation. In Fig. 7.3a, the main 5G (also referred to as IMT-2020) KPIs are summarized and compared to the 4G one (also known as IMT-Advance). In Fig. 7.3b, the three slices are compared highlighting their KPI focus.

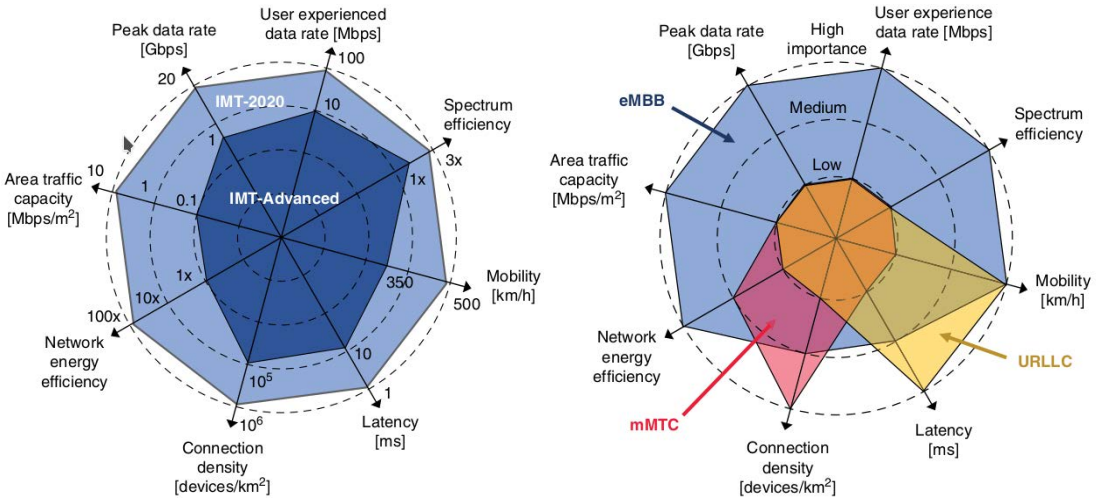


Figure 7.3: (a) Comparison between IMT-Advanced, also known as 4G, and IMT-2020, also known as 5G; (b) slices requirements comparison. [35]

7.1.4 URLLC technology components

The URLLC technology components can be classified into two main categories: the first related to latency reduction and the second to reliability enhancement.

The latency related technologies are the following:

- A new flexible frame structure is introduced to shorten the Transmission Time Interval (TTI), which is the time of transmission over the radio link. Furthermore, the number of OFDM symbols per TTI depends can be reduced from a time slot of 14 symbols to mini slots of down to 2 symbols.
- The idea of efficiently multiplexing URLLC and eMBB will be one key solution, if the amount of traffic is too high, the URLLC data is overwritten on eMBB slots ignoring the scheduling.
- Uplink grant-free transmission scheme allows data transmission without resource request, so that regular handshake delays, as sending the scheduling request and waiting for the uplink grant allocation are avoided.

Regarding the reliability, the technologies involved are:

- Micro-diversity, referring to having multiple antennas at either the transmitter side or the receiver side or both. For the purposes of URLLC performance evaluation, 4x4 was selected to get sufficient diversity orders.
- Several method for improving the control channels are provided, e.g. the adaptive configuration of Channel Quality Information report resource and negative acknowledgments protection.
- The Hybrid Automatic Repeat Request retransmissions add significant delay due to its high complexity. Proactive repetition considers fixed allocation of resources for K times transmissions to achieve reliability without substantial latency increase.

7.2 IEEE802

7.2.1 IEEE802.11AX

The IEEE802.11 standard evolution has been showing in the years an increasing nominal data rates: starting from 2 Mbps of 802.11 1997, reaching the 600 Mbps of 802.11n and the above Gbps rates of the 802.11ac. These rates were obtained with faster modulations, new coding schemes, wider channels and employing Multiple Input Multiple Output (MIMO).

However, the latest analyses showed that a further improvement of WiFi throughput needs innovative solutions. In particular, instead of widening the band or increasing the amount of spatial streams, it is better to adopt new channel access approaches. IEEE8021.11ax, or WiFi 6, is currently being designed following this philosophy.

Nowadays, many WiFi devices are deployed in the so called dense environments, that is spaces characterized by a huge amount of devices concentrated in very populated areas. Therefore, the main source of performance degradation is the massive interference among them.

In the past, efforts were devoted to decrease interference and reduce collisions by avoiding hidden stations, they are stations (STA) connected to the access point (AP) but unreachable for the other STAs. IEEE802.11ax utilizes a different approach in which all the STAs are exploited to enhance spatial reuse, i.e., simultaneous transmissions on overlapping networks to increase the final throughput.

IEEE802.11ax aims to improve the transmission of small packets, a problem that has never been resolved in the various amendments. First, they suffer from big overheads, and then, even if they are aggregated, a toll in terms of time for accessing the channel, separating frames and sending an acknowledgment, has to be paid.

Another challenge comes from the great amount of user-generated multimedia contents with consequent demand for uplink bandwidth. IEEE 802.11ax addresses it by extending the downlink (DL) Multi-User (MU) MIMO technology introduced in IEEE 802.11ac to uplink (UL).

IEEE802.11ax amendment was planned to be released in the first part of 2019, but since the

final consensus has not been reached yet, the release date is postponed to mid-2020.

Main features

In IEEE802.11ax, a new PHY layer with higher modulation and coding schemes is introduced. WiFi 6 does not increase the number of the MIMO spatial streams (still 8) and does not widen the channel. Subsequently, the nominal data rate is increased only to 9.6 Gbps.

One of the main features of IEEE802.11ax is the Orthogonal Frequency Division Multiple Access. It is an approach already used in cellular networks, but completely new for WiFi, aiming at scheduling in a more organized way the resources for each STA.

In general, the very wide channels in IEEE802.11ac, like 80 MHz, 80+80 MHz and 160 MHz, suffer from frequency selective interference. With OFDMA, adjacent subcarriers (tones) are grouped together into a Resource Unit (RU) and a sender can choose the best RU for each particular receiver, which actually results in higher Signal-to-Interference-plus-Noise Ratio (SINR), Modulation and Coding Scheme (MCS) and throughput.

In addition to all these advantages, it must be said that OFDMA is a more deterministic as approach for the medium access and it could support real-time communication for industrial applications. Therefore, it could theoretically approximate TDMA, even though it has not been fully demonstrated yet.

Some other features, completely innovative or taken from IEEE802.11ah, are implemented to enhance the reliability of the communication, as BSS color or two NAVs, and to diminish the power consumption, for example with Target Wake Time (TWT).

Modulation

The IEEE802.11ax PHY inherits many aspects from IEEE802.11ac. It is based on Orthogonal Frequency-Division Multiplexing (OFDM) and supports operations in 20 MHz, 40 MHz, 80 MHz, 80+80 MHz and 160 MHz channels. In oder to increase the tones, the duration of the OFDM symbols used for the PHY payload is incremented to 12.8 μs. Longer symbols allow to reduce the overhead due to Guard Intervals (GI) and they are more resilient to the inter-user jitter, which is beneficial for UL MU transmission. Moreover, different GI can be

selected (0.8 μs, 1.6 μs and 3.2 μs), which allows the reduction of overhead.

The 802.11ax amendment also introduces a new modulation techniques, 1024-QAM, which may be exploited in indoor scenarios with very good channel conditions. It offers a nominal data rate of 9.6 Gbps.

OFDMA

All the IEEE802.11 standards are client-centric and exploit a randomized, contention-based approach. Conversely, OFDMA is AP-centric and makes possible for a WiFi 6 AP to simultaneously communicate with multiple devices without waiting that the channel is completely free.

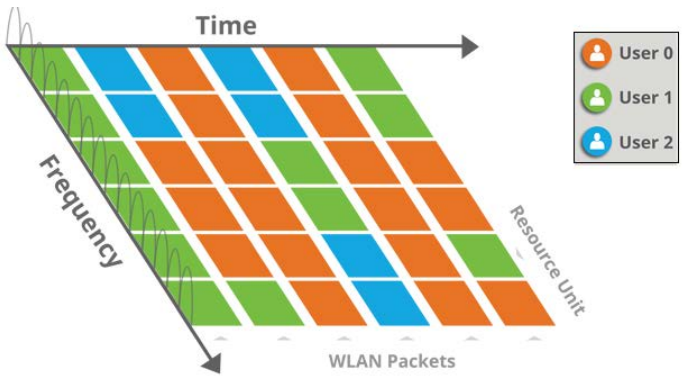


Figure 7.4: OFDMA example. [39]

In particular, the channel is split into sub-channels and it is allocated to the users, creating a grid of fundamental block, called Resource Units. It is depicted in Fig. 7.4.

OFDMA makes Wi-Fi radio access closer to LTE. For the 3GPP technology, OFDMA is time-based, many tones are associated to different user equipment during one transmission time interval. Instead in IEEE802.11ax, OFDMA is frame-based, which means that an MU frame comprehends data from and to different users, where multiple tones can be allocated to the users for the entire frame duration. An example is given in Fig. 7.5.

It must be said that OFDMA is built over the legacy Distributed coordination function (DCF) to retain compatibility with older amendments as well as guarantee a fair use of the

channel, and then OFDMA is managed by the AP. This means that the channel is still accessed with CSMA/CA, allowing to start an OFDMA frame only when it is free. It is considered busy in the following cases:

1 if a STA senses a frame preamble, the channel is considered busy for the duration specified in the preamble (physical sensing);

2 if a STA detects an unknown signal and the total power is 20 dBm above the minimum sensitivity level;

3 if Network Allocation Vector (NAV) is set to 1, the channel is accounted as busy (virtual sensing).

If the channel is found in one of these states, then the STA will back off for a random amount of time.

Hence it is clear that IEEE802.11ax is not fully deterministic because a random step is still necessary. Only after having accessed the channel, the frame is scheduled in a more deterministic way for the transmission: the AP can start an usual DL transmission, DL MU transmission with OFDMA, MIMO or both, or allocate RUs for UL MU transmission.

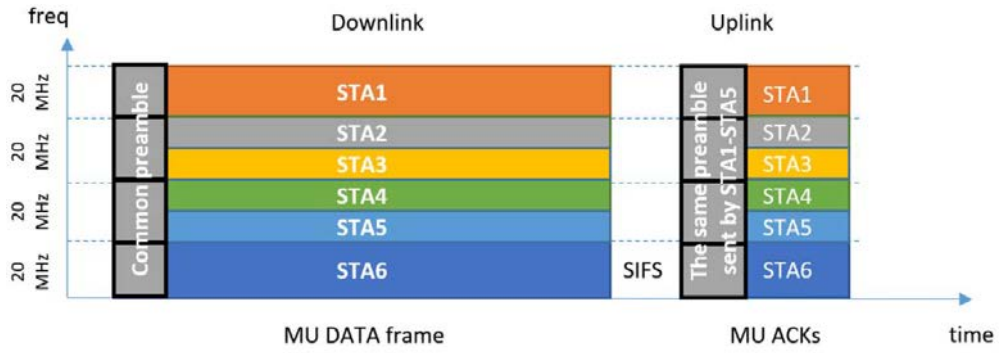


Figure 7.5: Example of OFDMA transmission in IEEE802.11ax. [40]

Moreover, this new approach improves the way to handle the overhead. For a DL MU transmission, the same PHY preamble is defined and used for every subchannel of the exploited channel, as presented in Fig. 7.5. It is allocated at the beginning of the frame and it specifies the frame duration and the tone mapping among STAs.

Conversely, for an UL MU transmission, the preamble from the preceding frame is reused,

therefore replies are transmitted over the same subchannels.

The UL sending starts exactly one Short InterFrame Space (SIFS) after the DL frame. This permits to synchronize the STAs participating in the UL MU transmission, whatever techniques the STAs use: OFDMA, MU-MIMO, or both. For an UL transmission may take more than a SIFS for a STA to prepare a UL transmission, subsequently, the AP can pad the Trigger frame, a new control frame for allocating the channel for UL. It is shown in Fig. 7.6.

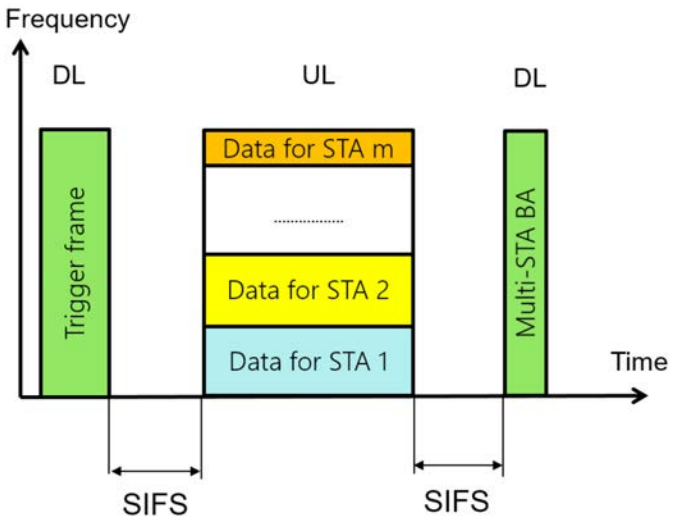


Figure 7.6: An example of UL OFDMA transmission. [40]

It is noteworthy that thanks to MU-MIMO, up to eight users can be assigned to an RU and it is also possible to allocate up to four spatial streams per user.

BSS color and two NAVs

The Basic Service Set (BSS) is a collection of STAs which may or may not include an AP. Accounting the dense environments in which these BSSs are located, different BSS can overlap with each other. It is useful to distinguish from which BSS a frame comes from, without decoding it entirely.

IEEE802.11ax assigns a non-unique 6 bits ID for the BSS, called BSS color, placed in the frame preamble. This fast identification of a BSS is used for determining channel access rules and for power saving.

Since the BSS color is selected randomly by the AP, the colors of two neighboring BSSs may coincide. In that case, the involved STAs warn the AP which starts the procedure of changing its BSS color. The new BSS color will be broadcast by means of beacons.

The WiFi 6 performs physical and virtual carrier sensing before transmitting a frame. For the virtual case, in the MAC header, a STA indicates the NAV value, i.e., for how long the following frame exchange will occupy the channel. The other STAs, after sensing this frame, will set NAV equal to 1 for the indicated time. The STA cancels its NAV state only when it receives an end frame.

Let's suppose now a STA with NAV equal to 1 and it receives an end frame from an Overlapping BSS (OBSS). The STA will reset the NAV and the medium will not be considered virtually busy anymore. So it could start its own transmission which may causes a collision. To prevent this situation, IEEE802.11ax STAs will support two NAVs: one for its own BSS and the other for all the OBSSs, and they will modify the NAVs separately.

TWT

IEEE802.11ax introduces the Target Wake Time mechanism to minimize the contention between STAs and to reduce power consumption. TWT allows a STA to periodically negotiate with an other STA or AP, when it can wake up for a certain amount of time (called TWT Service Period or TWT SP) and exchange frames.

The STA who has requested the TWT can always sleep, except during the TWT SP intervals. The STA is not required to wake up even for beacons, which can significantly improve the energy saving.

A full synchronization of TWT SPs among STAs is not conceived in the standard. Besides, during an established TWT SP, other STAs are not stopped from accessing the channel. So, TWT does not provide contention-free channel access, because still the channel must be sensed, but it helps to organize the transmissions.

7.2.2 IEEE802.11be

Even though IEEE802.11ax has not been released yet, the WiFi community is already preparing the next step with IEEE802.11be. A task group has been formed to select features and objectives, which are discussed in [41]. The first standard draft of this technology is planned for May 2021.

The main goals that this amendment wants to achieve are:

- throughput at least of 30 Gbps, using carrier frequencies between 1 and 7.125 GHz;
- backward compatibility with legacy 802.11 devices in the 2.4 and 5 GHz;
- implement a mode with improved worst case latency and jitter.

Regarding the last point, it is important to say that no values have been defined so far.

The most innovative peculiarity of this amendment is that it opens up the use of unlicensed spectrum between 5.925 and 7.125 GHz, which more than doubles the available bandwidth in the 5 GHz band. The task group proposes a channel size of 40 MHz up to 320 MHz.

Moreover IEEE802.11be targets to upgrade the spatial multiplexing up to 16 spatial streams. This can double the spectral efficiency with respect to IEEE802.11ax, provide a better integration between WiFi and the home fiber connection and also improve the use of rich scattering in the indoor environments.

Consequently, this larger exploitation of the spectrum and the increase of the spatial streams request new ideas to improve the efficiency. From a industrial perspective, this amendment shows some promising features towards real-time communication, with various solutions aiming at reducing latency, and improving determinism.

The principal features selected by the task group are discussed in the following.

Multi-band/multi-channel aggregation and operation

Dual-radio STAs and tri-band APs will emerge, capable of operating at 2.4, 5, and 6 GHz at the same time. An efficient use of these multiple bands is for IEEE802.11be one of the principal tasks and different ideas, which are still under evaluation, are proposed.

- Multi-band data aggregation. By aggregating the 5 and 6 GHz spectrum, higher throughput peaks for data transmission and reception can be obtained.
- Simultaneous transmission and reception in different bands/channels. Using simultaneous uplink/downlink operations in separate bands/channels can reduce interference and latency and can enhance throughput.
- Simultaneous transmission and reception in the same channel. Some analyses operated by the task group concluded that full duplex operations can be realized with little modifications from the 802.11ax standard.
- Data and control plane separation. By dedicating a band/channel to data transmission/reception and a complementary one to provide frequent and reliable control information updates, delays for management and scheduling can be removed.

Access point coordination

Enabling some degree of collaboration among neighboring 802.11be APs will permit a more efficient utilization of the limited time, frequency, and spatial resources available. The main proposed solutions are here expressed.

- Coordinated OFDMA. APs have to synchronize their data transmissions in order to avoid collisions and waste of time. This is a significant step towards getting a more deterministic solution, indeed a full synchronization would remove the randomness introduced by contention-based channel access methods.
- Coordinated Null Steering. Multi-antenna APs provide high signal power gains through beamforming. APs can also leverage their antennas to emit null spatial radiation toward OBSS, this approach is referred to as coordinated null steering or coordinated beamforming.

7.3 Wireless TSN

Deterministic and real-time communications are a clear need in automotive, industrial automation and audio/video sectors. The main wired industrial networks, which achieve these features, are realized quite often with semiproprietary fieldbus communication, as ProfiBUS, EtherCAT, CANbus, etc. Most of these technologies have in common that they are based on the Ethernet standard IEEE802.3, but they effectively act as closed and non-compatible systems.

The Institute of Electronics and Electrical Engineers, the Internet Engineering Task Force and the International Electrotechincal Commission decided to address these features and improving deterministic and real-time networking directly in the Ethernet standard. The IEEE Time-Sensitive Networking (TSN) task group, previously called the IEEE Audio/Video Bridging, has been working on proposing new standards and redefining old ones to provide an open and standardized technology, not affiliated to any organization or company, which satisfies the interoperatibility, determinism and real-time requirements.

Although TSN is not yet mature, it is investigated how it is possible to bridge it with the various wireless solutions without losing its peculiarities. It is well known that wireless networks add many advantages, like an increased flexibility and mobility of devices, better cost efficiency and at the same time reduced complexity. On the other hand, wireless links introduce unreliability, interference and latency, making that challenging to guarantee the performance.

The TSN task group is including inside some the standards, some features in order to ease the integration between TSN and wireless networks. Furthermore, most of the TSN standards are rather generic regarding the specification on the layer 2 protocol. This permits more flexibility when trying to extend TSN to a wireless technology like any IEEE802.11 amendment or the upcoming 5G.

The main key features of TSN are briefly discussed and then a brief description of research towards maintaining these features in bridged wired-wireless networks is presented.

7.3.1 Time sensitive networking

TSN fundamental characteristics are provided by four key concepts and for each of them the task group has defined multiple standards to realize the necessary functionality. They are summarized in Fig 7.7.

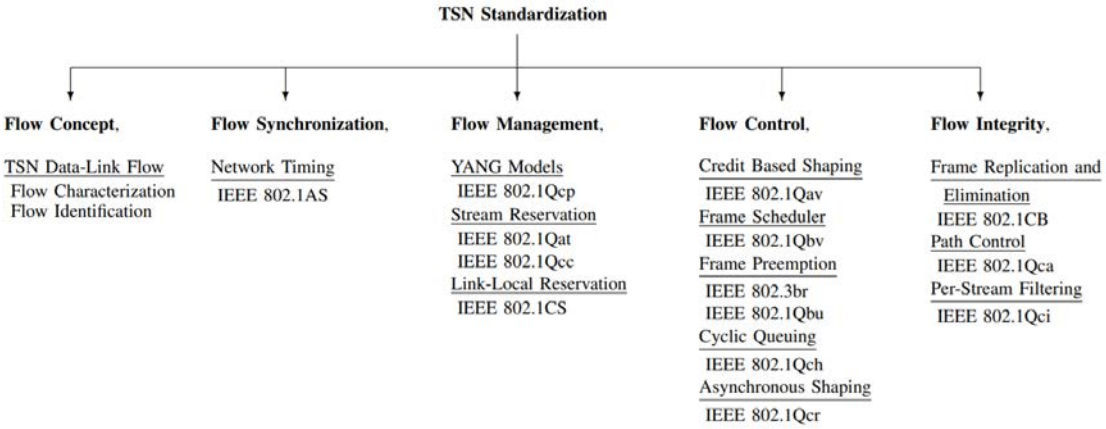


Figure 7.7: Classification of TSN standardization. [42]

A special look is given to synchronization because it is the standard on which most of the others standards are based on and also it is crucial for obtaining real-time awareness.

Time Synchronization

The first aspect that TSN aims to achieve is an accurate time synchronization among all nodes in the network and this is provided by IEEE802.1as by means of the evolution of Precision Time Protocol (PTP), called generalized PTP (gPTP). This synchronicity is crucial in order to ensure real-time networking and for the other involved standards.

PTP is currently used to synchronize all the integrated clocks in network interface cards (NICs) with a master clock, with an accuracy below one microsecond.

The gPTP synchronizes device clocks in the network exchanging relevant time event messages via UDP. These messages, which travel through the Clock Master (CM) and the Clock Slaves (CS), define a time-aware network, also called gPTP domain, as shown in Fig. 7.8. The network is organized as a spanning tree structure with the GrandMaster (GM) clock at the root of the hierarchy. The most accurate clock source is chosen as GM by the Best Mas-

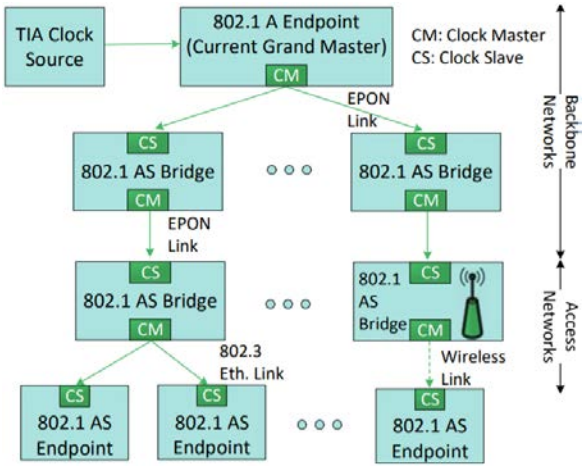


Figure 7.8: gPTP domain in IEEE802.11as. [42]

ter Clock Algorithm; usually the GM is synchronized with the International Atomic Time (TIA).

To create a time-aware network, two parameters are involved: the residence time in each node and the link latency for each link. The former accounts for the ingress and egress processing, queuing, and transmission time, the latter, instead, for the propagation delay between adjacent nodes with respect to the GrandMaster (GM).

For example, in Fig. 7.8, the bottom left endpoint receives time information from the upstream CM including the cumulative time from the GM. The device computes the path delay between the local CS and the direct CM to correct the received time. After correcting the received time, the endpoint should be fully synchronized with the gPTP GM clock.

Bounded Low Latency

TSN provides many standards for obtaining low latency, as 802.1Qbv for Time Scheduled Traffic together with IEEE802.1av, also called Credit Based Shaper (CBS).

802.1Qbv creates a cyclic time scheduling, where, time is split into labeled slots in which only specific traffic classes can be forwarded. The CBS divides data into two classes, the tight low bound and loose delay bound class, and stores them in two queues.

Moreover, the CBS manages the transmission process, by assigning to the queue a certain amount of credits when an entry is ready and the channel is not busy. Then a queue can spend one credit to send each data entry.

Ultra Reliability

TSN provides the standard 802.1CB for Seamless Redundancy and Identification for streams. It replicates frames and sends them to the recipient through at least two different paths. The extra frames will be eliminated on the last network node before the target.

This is realized in collaboration with 802.1Qca for Path Control and Reservation, which determines and reserves the available paths in the network.

Resource Management

The last key concept of TSN is the configuration and management of all the available network resources. TSN standard 802.1Qcc defines a fully centralized or a fully distributed model for network configuration.

The main option is the centralized one, it comprehends a Centralized User Configuration (CUC) and a Centralized Network Configuration (CNC) system. It works that an application requests the resource at the CUC, which triggers the appropriate actions to be taken at the CNC for resource reservation, like for configuring the cycle scheduling fin 802.1Qbv.

The Resource Management part of TSN is one of the parts which is still undergoing development and standardization.

7.3.2 Challenges and gaps for wireless TSN

Even though 802.11 and 3GPP standards have spent a lot of efforts to address latency and reliability requirements, it is difficult to realize a full integration of these technologies with a TSN.

It can be seen in Fig. 7.9 that TSN capabilities are performed at layer 2, specifically at the LLC. Since Ethernet (IEEE802.3) is the current media supporting TSN and IEEE802.11 is natively an IEEE802 technology, they share several capabilities that could make the integration seamless.

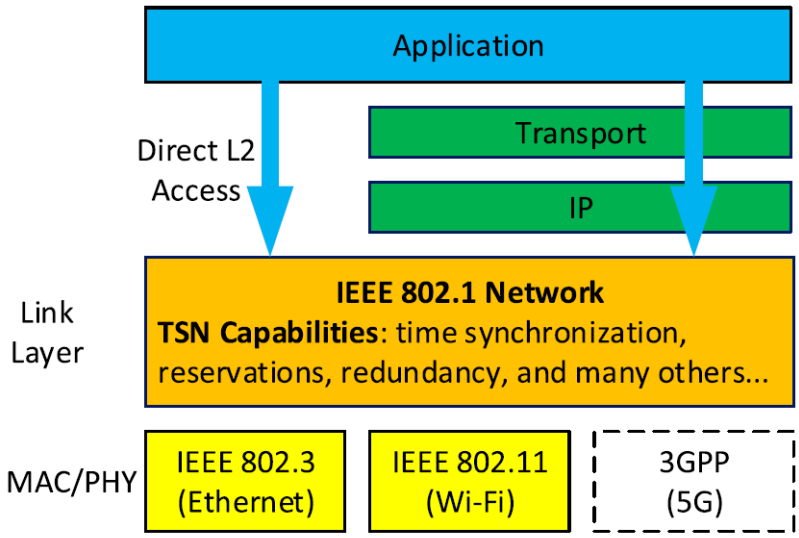


Figure 7.9: TSN protocol stack. [43]

For example, TSN IEEE802.1AS time synchronization is already supported in 802.11. The next step for 802.11 is the support for additional TSN capabilities, such as time-aware shaping, redundancy, and preemption which are, however, more challenging to implement over wireless.

Regarding 3GPP, the first common layer that 4G and 5G networks share with IEEE802-based networks is the IP layer. Hence standardization activities are necessary to integrate 5G networks with TSN networks. This problem has been already identified and consensus way to solve it is to introduce some translators in the 5G architecture.

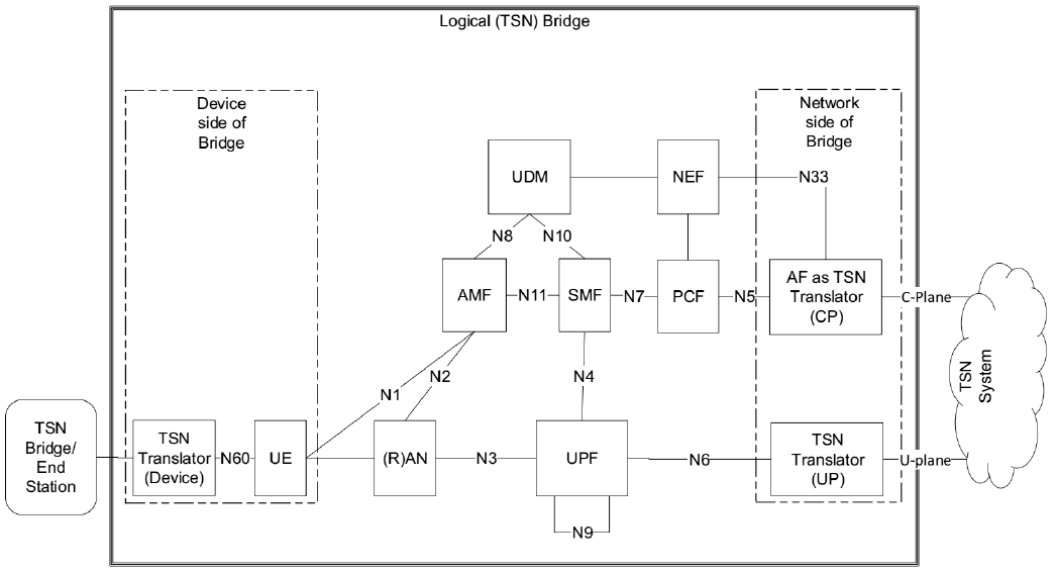


Figure 7.10: Integrating TSN in 5G architecture. [44]

As it is possible to see in fig. 7.10, devices able to map to/from 5G QoS framework and to maintain synchronization are placed in entrance positions to access the 5G architecture.

Regardless of the wireless communication adopted for integrating TSN, a roadmap has been outlined in [45] involving 5 step:

1 wireless configuration of wired TSN;

2 hybrid wired-wireless time synchronization;

3 wireless TSN scheduling;

4 wireless redundancy for wired TSN;

5 wireless TSN switch deployment.

The first phase concerns the use of wireless links to configure a wired TSN and takes place before that the system becomes operational. This is confined to actions taken by the Centralized Network Configurator (CNC) and the Centralized User Configuration (CUC) processes, including configuration of TSN schedules. It is accounted as straightforward because no technical or standardization gaps are here identified.

The second phase consists in integrating wireless gPTP with wired gPTP because the wireless section of the network has to rely on completely reliable time synchronization as the wired one. Fine Timing Measurement (FTM) defined in the IEEE802.1AS is expected to be the definitive method to provide synchronization. Main questions for this phase are if the wireless and the wired devices can synchronize accurately enough and if the introduction of Global Navigation Satellite System (GNSS) could be a feasible way to reach the goal.

Data scheduling for obtaining deterministic traffic is a necessary requirement for industrial systems, therefore wireless TSN must support IEEE802.1Qbv. The third phase regards the IEEE802.1Qbv scheduling implementation in wireless TSN, which alongside a complete synchronization, should eliminate frame collisions on the network.
Since surrounding networks may cause interference affecting determinism, the presence of OBSS and other sources of electromagnetic interference can be defined as a function of bit error rate.
There are still many unsolved questions like if it is possible to implement IEEE802.1Qbv on a wireless network and what happens if there is a synchronization error.

In the fourth phase, a deployment of the wireless system takes place in the form of increasing reliability by adding redundant communication channels. It is important to say that TSN will be introduced gradually into industrial systems and IEEE802.1CB is the standard for wired TSN reliability. It is still not clear if the redundancy will overload the wireless network or cause packets to be delivered out of order.

In the fifth and last phase, wireless TSN is deployed among the switches, that means that wireless communication is now deployed anywhere within a wired TSN network. At this stage, a full integration of wireless TSN that meets wired TSN reliability and performance requirements should be ready.

"Omai convien che tu così ti spoltre", disse 'l maestro; "ché,
seggendo in piuma, in fama non si vien, né sotto coltre".

Dante, Inferno, XXIV, 45-48

8 Conclusion

In this thesis various possibilities towards a better introduction of wireless networks in industrial use cases, particularly in mobile robotics, have been investigated.

Starting from the main uses cases, the necessary communication requirements have been outlined and a comparison among the most common solution has been carried out. It is clear that no technology fully satisfies the strict industrial demands in terms of reliability, low jitter, determinism and real-time, and consequently, wired solutions are still preferred to the wireless one. However, a recently standardized technology, called WIA-FA, seems promising, therefore its peculiarities have been discussed and tested in different experiments.

In this thesis, two different experimental applications of wireless technologies for industrial purposes have been performed: one regarding the critical motion control in robotics and one concerning the integration between a fieldbus communication and wireless technology. What can be found in the literature concerning these two topics are only theoretical studies and only few papers are present for both cases. Therefore, the experiments presented in this thesis constitute a novel result.

In the first application, different communication technologies have been tested to verify which one can better support EGM, the ABB interface to control robots from external devices.

LTE is reliable, proved to be a reliable technology whose latency (in the order of tens of ms), however, does not make it a good candidate for EGM.
Tests performed with IEEE802.11n demonstrated that it also cannot be considered a reliable solution due to its high RTT values lack of determinism.
WIA-FA, instead, seems to perform very similar to Ethernet in terms of both mean and variation of RTT. No packet losses occurred for both of them in the tests, carried out with short distance and line of sight.
A final test performed in this application concerns layer 2 mode simulation, in which an EGM communication is replicated using packet at layer 2 rather than UDP ones, as defined in the protocol. The test showed impressive results, with the average RTT values dropping under one millisecond for both Ethernet and WIA-FA, still without no packet losses in the communications.

The second case is a preliminary experimental study on the feasibility of integrating one of the most common fieldbus communication technologies, CANbus, with WIA-FA in an application used for the motion control of prototype mobile robotics platforms.
WIA-FA confirmed to be a reliable wireless technology that can operate a bridging with CANbus, working at 2 ms refresh rate.
During the tests, it was clear that the CANbus packet generation scheme was not perfectly matching the WIA-FA characteristics. Hence, several solutions have been proposed to improve the results of the integration.
In particular, an adaptive solution, which exploited a multithread software to handle different flows separately, obtained an overall packet loss of 1.68% in a 180 seconds communication.
The experiment is concluded by emulating the CANbus packet generation on the same board that handles the WIA-FA traffic. It showed that a complete synchronization between the two traffic flows could lead to better results since the overall loss rate in a 180 seconds communication has been reduced to 0.003%.

After all these assessments, it can be said that WIA-FA is a promising solution because it provides a deterministic MAC and several features to improve the communication robustness in industrial uses cases. Also the tests confirmed that WIA-FA can be exploited to create an hybrid network with CANbus as long as a good synchronization between the parties is realized.

The thesis is concluded discussing the most interesting future releases of the major wireless

technologies, highlighting the innovative features that should provide better solutions towards the strict industrial requirements.

8.1 Future work

Although good results have been shown by the experimental studies, it must be said that such tests were performed in a point to point fashion, at a short distance, in line of sight and in a laboratory environment. The next step consists hence in replicating all the assessments in a more realistic testbed with more nodes.

Here is however a high level of confidence regarding the WIA-FA potentiality in these scenarios, since some tests in spaces similar to an industrial environment have been already performed, with different applications purposes, showing no significant variation in packet losses and RTT.

Concerning the critical motion application, first future step is to perform also the RWS handshake over wireless, then more trajectories must be investigated. Moreover, position error results need to be discussed with motion control experts to clarify the meaning of the various peaks.

Regarding the bridging application, the adaptive solution has still an improvement margin by using a more accurate way to manage time. More CAN slaves can be added to assess the limits of WIA-FA and finally, giving the good performance, the hybrid setup can be moved to a real mobile robot, keeping the motion controller separate and communicating over wireless with the drive units.

A

Circuit schematics and wiring

The F3 is equipped with an integrated CAN transceiver and its circuit schematic is reported in Fig. A.1. The CAN transceiver is labeled as U1 and the output pins as J1.

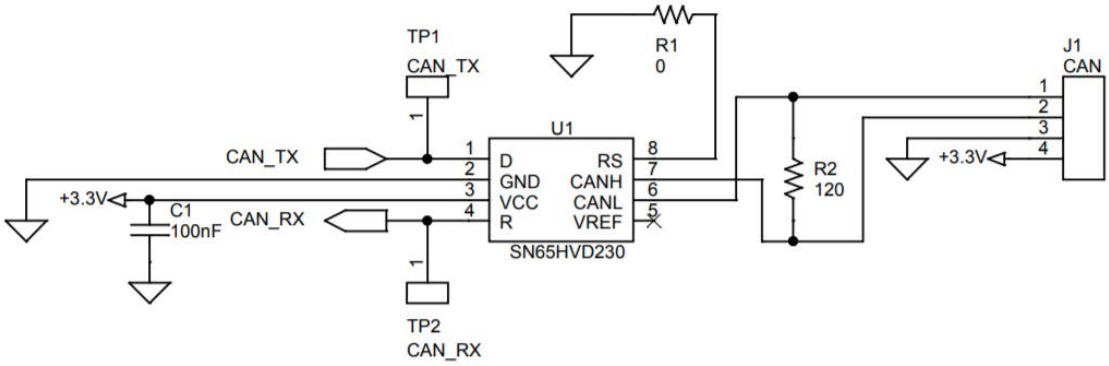


Figure A.1: F3 circuit schematic. [28]

Messages are transmitted changing the voltage level of the bus and these changes are elaborated by the transceiver using CAN High (CANH) and CAN Low (CANL) outputs. Whenever the controller needs to send a message, it is forwarded to the D port of U1, instead when a packet is received it will be output by U1 at port R.

Moreover, it is important to notice that a resistor (R2) is located between the CANH and CANL outputs, to terminate this end of the circuit.

CANL and CANH pins from J1 are wired to pins 3 and 4 of a female D-sub 9 connector, presented in Fig. A.2b (bottom). All the other remaining D-sub 9 pins are connected to the UART output, for flashing, and to the SWD output, for setting controller's parameters like ID number. This is depicted in the rightmost part of Fig. A.3.

(a) (b)

Figure A.2: (a) Transceiver and IC adapter. (b) DSUB9 port.

The master controller internal schematics are not shown here due to their complexity, but they can be found in [29]. In the F7, the CAN transceiver is not integrated in the controller, therefore an external one must be adopted. The same chip SN65HVD230 used in the F3 and shown in Fig. A.1 as U1 is adopted as an external CAN transceiver in the F7. To ease the installation, the transceiver is soldered to an IC adapter, RE899. The two components are depicted in Fig. A.2a.

The R and D output of the CAN transceiver are respectively wired to F7's outputs PB8 and PB9. Then, two more F7 outputs are exploited as power input (V_{cc}) and ground (GND) to power supply the transceiver.

It is common practice when handling an external CAN transceiver to connect CANL and CANH, respectively, to pin 2 and 7 of a female D-sub 9 connector, as shown in the leftmost part of Fig. A.3. Moreover, for F7, a 120 Ω resistor is located between pin 2 and 7 of the D-sub 9 to terminate the circuit according to the the CAN requirements.

The Zynq board is not equipped with an integrated CAN transceiver, therefore the one shown in Fig. A.2a is used also for this board. The J58 port has been selected as I/O port for the CANbus and it is depicted in Fig. A.4. Only pins 6, 8, 10 and 12 are used, in particular

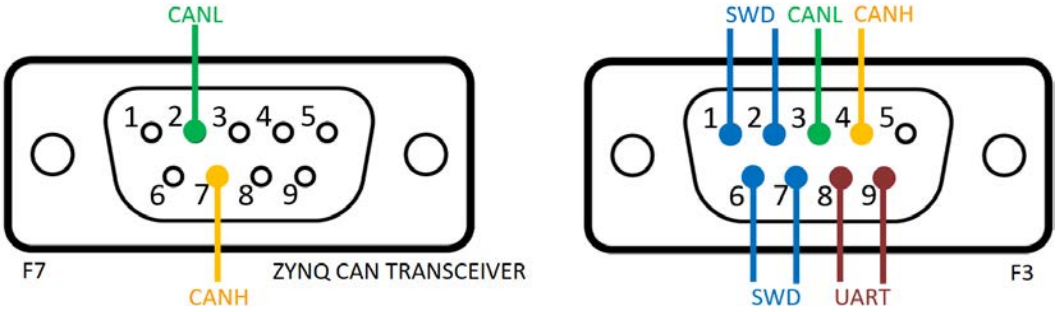


Figure A.3: Female D-sub 9 wiring for F7 and ZYNQ CAN transceiver on the left and for F3 on the right.

6 is adopted for receiving, R, and 8 for transmitting, D.

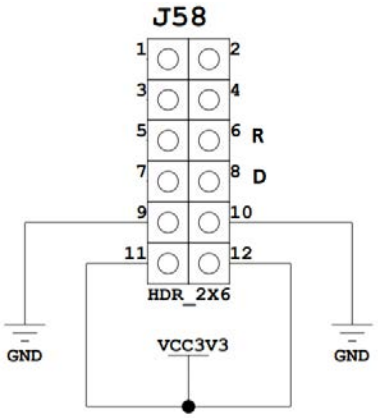


Figure A.4: CAN pin outputs.

The R, D, V_{cc} and GND pins in J58 are respectively wired to the corresponding transceiver outputs. The CANL and CANH of the transceiver are connected to pin 2 and 7 of the female D-sub 9, as shown in the leftmost part of Fig. A.3; besides a 120 Ω resistor is soldered between the pins because the end must be terminated.

A bus is needed to connect a controller to the platform and it differs depending on if it will be used for F7 or F3. For the former, it is made by simply wiring two male D-sub 9 connectors from pin 2 to the other pin 2 and the same way for pins 7, as visible in Fig. A.5. Also for the the latter, that is presented in the leftmost part of Fig. A.6, two male D-sub 9 connectors are involved: for CANL, at the F3 end a wire is attached to pin 3 and at the board end to pin 2; for CANH, at F3 end pin 4 is used and it is linked to pin 7 at the board end.
When multiple slave are added in the architecture, it is to necessary to modify the CANbus:

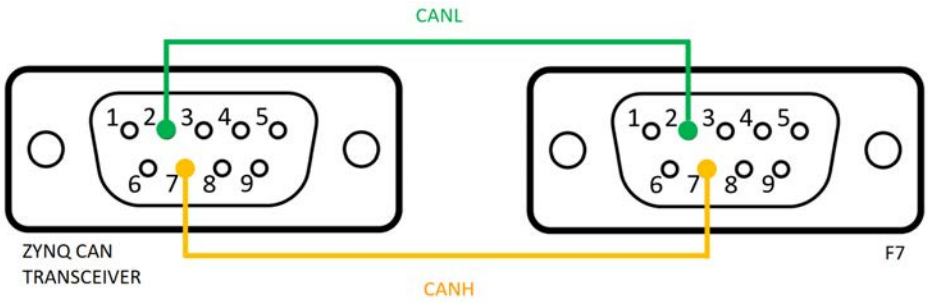


Figure A.5: Male D-sub 9 wiring for the bus between F7 and ZYNQ CAN transceiver.

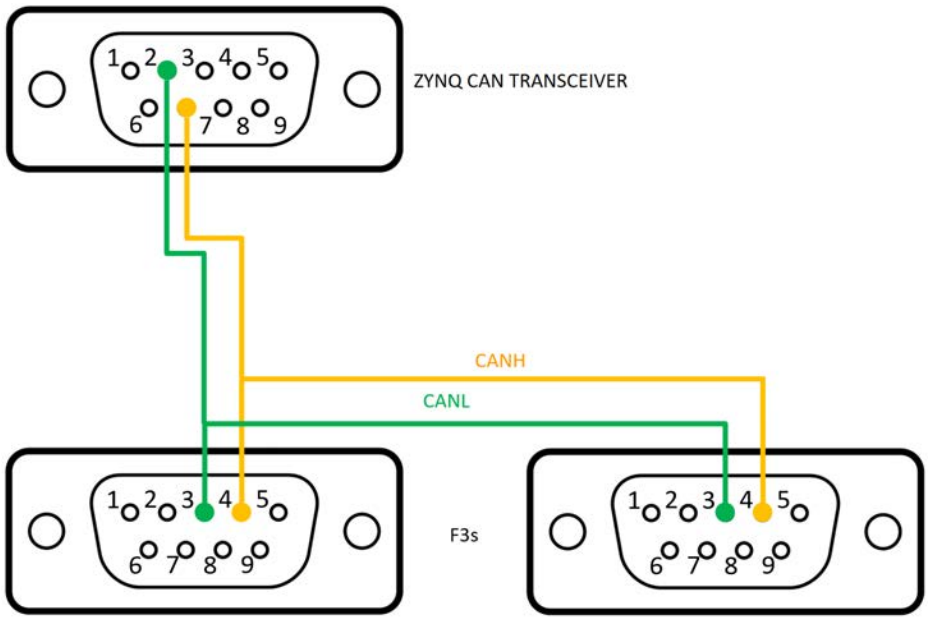


Figure A.6: Male D-sub 9 wiring for the bus among two F3s and ZYNQ CAN transceiver.

more male D-sub 9 connectors, one for each new slave, must be connected to one of the end of the old bus by soldering new wires from pin 3 to pin 3 and the same for pins 4. A generic example for a bus with only two F3s is depicted in Fig. A.6. Moreover it must be remembered that only the first and the last controller must be terminated to avoid reflections.

References

[1] ABB, "ABB demonstrates concept of mobile laboratory robot for Hospital of the Future," https://new.abb.com/news/detail/37301/abb-demonstrates-concept-of-mobile-laboratory-robot-for-hospital-of-the-future, 2019.

[2] J. Ke, "Five Critical Elements of Uninterrupted Wireless Connectivity for AS/RS and AGV Systems," White paper, February 2016.

[3] S. G. Tzafestas, *Introduction to mobile robot control*, 1st ed. Elsevier, 2014.

[4] C. Savant, "Design of Driveline for Mobile Robot Platform," Master's Thesis, KTH, Industrial Engineering and Management, 2018.

[5] M. Schneier, M. Schneier, and R. Bostelman, *Literature review of mobile robots for manufacturing*. US Department of Commerce, National Institute of Standards and Technology, 2015.

[6] *Application manual - Controller software IRC5*, 4th ed., ABB AB, 2016.

[7] R. Zurawski, *Industrial communication technology handbook*. CRC Press, 2014, ch. 52.

[8] C. Alliance, "What is CBRS?" https://www.cbrsalliance.org/resource/what-is-cbrs/, 2018.

[9] Moxa, "Smart wireless sends warehouses into smart territory," https://www.moxa.com/en/case-studies/smart-wireless-sends-warehouses-into-smart-territory, 2016.

[10] A. A. Tabassam, H. Trsek, S. Heiss, and J. Jasperneite, "Fast and seamless handover for secure mobile industrial applications with 802.11r," in *2009 IEEE 34th Conference on Local Computer Networks*. IEEE, 2009, pp. 750–757.

[11] M. Luvisotto, Z. Pang, and D. Dzung, "Ultra high performance wireless control for critical applications: challenges and directions," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1448–1459, 2016.

[12] P. Riihikallio, "How many users can one Wi-Fi access point support?" https://metis.fi/en/2018/02/how-many-users/, 2018.

[13] A. von Nagy, "Wi-Fi Roaming Analysis with Wireshark and AirPcap," https://www.revolutionwifi.net/revolutionwifi/2013/01/wi-fi-roaming-analysis-with-wireshark.html, 2013.

[14] F. Mazzenga, D. Cassioli, P. Loreti, and F. Vatalaro, "Evaluation of packet loss probability in bluetooth networks," in *2002 IEEE International Conference on Communications. Conference Proceedings. ICC 2002 (Cat. No. 02CH37333)*, vol. 1. ieee, 2002, pp. 313–317.

[15] C. Dombrowski and J. Gross, "EchoRing: a low-latency, reliable token-passing MAC protocol for wireless industrial networks," in *Proceedings of European Wireless 2015; 21th European Wireless Conference*. VDE, 2015, pp. 1–8.

[16] R3Communications, "Echoring™ calculator," https://echoring.com/calculator/.

[17] Bundesnetzagentur, "Key elements for the rollout of digital infrastructures and identification of demand for nationwide assignments in the 2 ghz and 3.6 ghz bands," 2017.

[18] H. Chen, R. Abbas, P. Cheng, M. Shirvanimoghaddam, W. Hardjawana, W. Bao, Y. Li, and B. Vucetic, "Ultra-reliable low latency cellular networks: use cases, challenges and approaches," *IEEE Communications Magazine*, vol. 56, no. 12, pp. 119–125, 2018.

[19] P. Skarin, W. Tärneberg, K.-E. Årzen, and M. Kihl, "Towards mission-critical control at the edge and over 5G," in *2018 IEEE International Conference on Edge Computing (EDGE)*. IEEE, 2018, pp. 50–57.

[20] G. Mester, "Wireless sensor-based control of mobile robots motion," in *2009 7th International Symposium on Intelligent Systems and Informatics*. IEEE, 2009, pp. 81–84.

[21] Y.-H. Wei, Q. Leng, S. Han, A. K. Mok, W. Zhang, and M. Tomizuka, "RT-WiFi: Real-time high-speed communication protocol for wireless cyber-physical control applications," in *2013 IEEE 34th Real-Time Systems Symposium*. IEEE, 2013, pp. 140–149.

[22] R. P. Gomes, J. E. Oliveira, and F. J. Cardoso, "Integrating Zigbee and CAN networks in industrial applications," in *2010 6th IEEE International Conference on Distributed Computing in Sensor Systems Workshops (DCOSSW)*. IEEE, 2010, pp. 1–2.

[23] W. Chen, L. Kong, Y. Wu, X. Liao, and C. Tao, "Wireless extensions of CANBUS in industrial applications," in *2011 4th International Conference on Biomedical Engineering and Informatics (BMEI)*, vol. 4. IEEE, 2011, pp. 2187–2191.

[24] X. Ren, C. Fu, T. Wang, and S. Jia, "CANbus network design based on bluetooth technology," in *2010 International Conference on Electrical and Control Engineering*. IEEE, 2010, pp. 560–564.

[25] N. Pereira, B. Andersson, and E. Tovar, "Implementation of a dominance protocol for wireless medium access," in *12th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA'06)*. IEEE, 2006, pp. 162–172.

[26] W. Liang, M. Zheng, J. Zhang, H. Shi, H. Yu, Y. Yang, S. Liu, W. Yang, and X. Zhao, "WIA-FA and its applications to digital factory: A wireless network solution for factory automation," *Proceedings of the IEEE*, vol. 107, no. 6, pp. 1053–1073, 2019.

[27] ABB, "abb_libegm," https://github.com/ros-industrial/abb_libegm, 2019.

[28] STMicroelectronics, "Electronic speed controller for BLDC and PMSM three phase brushless motor, User manual," https://www.st.com/content/ccc/resource/technical/document/user_manual/groupo/06/a3/c1/ae/7d/27/4c/e0/DM00384353/files/DM00384353.pdf/jcr:content/translations/en.DM00384353.pdf, 2018.

[29] ——, "STM32 Nucleo-144 boards, User manual," https://www.st.com/content/ccc/resource/technical/document/user_manual/groupo/26/49/90/2e/33/od/4a/da/DM00244518/files/DM00244518.pdf/jcr:content/translations/en.DM00244518.pdf, 2017.

[30] Xilinx, "linux-xlnx," https://github.com/Xilinx/linux-xlnx, 2019.

[31] ——, "PetaLinux Tools Documentation Reference Guide," https://www.xilinx.com/support/documentation/sw_manuals/xilinx2019_1/ug1144-petalinux-tools-reference-guide.pdf, 2019.

[32] "SocketCAN," https://github.com/linux-can/can-utils/, 2019.

[33] M. D. Phil Norman, Elliott Hughes, "Terminator," https://code.google.com/archive/p/jessies/wikis/Terminator.wiki, 2008.

[34] Z. Li, M. A. Uusitalo, H. Shariatmadari, and B. Singh, "5G URLLC: Design challenges and system concepts," in *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*. IEEE, 2018, pp. 1–6.

[35] A. Al-Dulaimi, X. Wang, and I. Chih-Lin, *5G Networks: Fundamental Requirements, Enabling Technologies, and Operations Management*. John Wiley & Sons, 2018.

[36] Open Networking Foundation, "SDN Architecture," *Issue 1.1*, vol. ONF TR-521, 2016.

[37] J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J. J. Ramos-Munoz, J. Lorca, and J. Folgueira, "Network slicing for 5g with SDN/NFV: concepts, architectures and challenges," *arXiv preprint arXiv:1703.04676*, 2017.

[38] A. Kaloxylos, "A survey and an analysis of network slicing in 5g networks," *IEEE Communications Standards Magazine*, vol. 2, no. 1, pp. 60–65, 2018.

[39] D. Huang, "802.11ax fundamentals: Orthogonal Frequency-Division Multiple Access," https://theruckusroom.ruckuswireless.com/wired-wireless/technologytrends/802-11ax-fundamentals-orthogonal-frequency-division-multiple-access/, 2018.

[40] E. Khorov, A. Kiryanov, A. Lyakhov, and G. Bianchi, "A tutorial on IEEE 802.11 ax high efficiency WLANs," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 197–216, 2018.

[41] D. López-Pérez, A. Garcia-Rodriguez, L. Galati-Giordano, M. Kasslin, and K. Doppler, "IEEE 802.11 be Extremely High Throughput: The Next Generation of

Wi-Fi Technology Beyond 802.11 ax," *IEEE Communications Magazine*, vol. 57, no. 9, pp. 113–119, 2019.

[42] A. Nasrallah, A. S. Thyagaturu, Z. Alharbi, C. Wang, X. Shao, M. Reisslein, and H. El-Bakoury, "Ultra-low latency (ULL) networks: The IEEE TSN and IETF DetNet standards and related 5G ULL research," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 88–145, 2018.

[43] D. Cavalcanti, J. Perez-Ramirez, M. M. Rashid, J. Fang, M. Galeev, and K. B. Stanton, "Extending accurate time distribution and timeliness capabilities over the air to enable future wireless industrial automation systems," *Proceedings of the IEEE*, vol. 107, no. 6, pp. 1132–1152, 2019.

[44] M. Sajadieh, "5G URLLC–The Path to Introduce High Performance Wireless for Industrial Networking," *TSN/A Conference*, 2019.

[45] S. F. Bush and G. Mantelet, "Industrial Wireless Time-Sensitive Networking: RFC on the Path Forward," https://avnu.org/wp-content/uploads/2014/05/Industrial-Wireless-TSN-Roadmap-v1.0.3-1.pdf, 2018.

[46] T. Sauter, "The three generations of field-level networks—Evolution and compatibility issues," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 11, pp. 3585–3595, 2010.

[47] I. E. Commission *et al.*, *Industrial Networks-Wireless Communication Network and Communication Profiles-WIA-FA*. International Electrotechnical Commission, 2017, no. IEC 62948.

[48] J. Åkerberg, M. Gidlund, and M. Björkman, "Future research challenges in wireless sensor and actuator networks targeting industrial automation," in *2011 9th IEEE International Conference on Industrial Informatics*. IEEE, 2011, pp. 410–415.

[49] M. Luvisotto, Z. Pang, and D. Dzung, "Ultra high performance wireless control for critical applications: challenges and directions," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1448–1459, 2016.

[50] T. Instrument, "SN65HVD23x 3.3-V CAN Bus Transceivers, Data sheet," http://www.ti.com/lit/ds/symlink/sn65hvd230.pdf, 2018.

[51] P. Marsch, Ö. Bulakci, O. Queseth, and M. Boldi, *5G system design: architectural and functional considerations and long term research*. John Wiley & Sons, 2018.

[52] N. ETSI, "Network functions virtualisation (nfv) release 3; evolution and ecosystem; report on network slicing support with etsi nfv architecture framework," 2017.

[53] ETSI, "3GPP TS 23.501 V15.2.0," 2018.

[54] G. P. A. W. Group, "View on 5G Architecture (Version 2.0)," 2017.

[55] B. Bellalta, "IEEE 802.11 ax: High-efficiency WLANs," *IEEE Wireless Communications*, vol. 23, no. 1, pp. 38–46, 2016.

[56] A. Mildner, "Time Sensitive Networking for Wireless Networks-A State of the Art Analysis," *Network*, vol. 33, 2019.

Acknowledgments

No choice could have been better than finalizing my master degree at ABB Corporate Research in Sweden. It is part of my attitude to never settle down, to look for something that allows me to make a qualitative step as a man as well as an engineer, something consistent with my high expectations. The only way I see it possible is by being surrounded by the best engineers and by being mentored by great chiefs, eager to share their experience. I have to admit that taking this choice was not easy, but now, at the end of this journey, it is clear what a fantastic learning adventure I have gone through. I must thank Michele Luvisotto to have believed in me by giving me this opportunity, he demonstrated to be a great supervisor, a talented researcher and an inspiring man. I would like to thank Zhibo Pang, a sharp engineer and also an incredible mentor, that allowed me to be part of this incredible team which aims only to high results while handling cutting edge technologies. I need to extended my gratitude to Prof. Stefano Vitturi, my supervisor at UNIPD, for his kindness and for being a solid support throughout this experience. A special thank is for my family, which must have found hard to let me go so far for so long, and for all my friends, that confirmed that true friendships are not hindered by long distances. Finally, a ”tack så mycket” is for all the beautiful people I have met here and left a sign forever.

Västerås, 11-11-2019