Mälardalen University Doctoral Dissertation 415

Dependable real-time communications for systems with integrated wired and wireless connectivity

Pablo Gutiérrez Peón

Mälardalen
University

Mälardalen University Press Dissertations
No. 415

DEPENDABLE REAL-TIME COMMUNICATIONS FOR SYSTEMS WITH INTEGRATED WIRED AND WIRELESS CONNECTIVITY

Pablo Gutiérrez Peón

2024

School of Innovation, Design and Engineering

ISBN 978-91-7485-673-6
ISSN 1651-4238
Printed by E-Print AB, Stockholm, Sweden

Mälardalen University Press Dissertations
No. 415

DEPENDABLE REAL-TIME COMMUNICATIONS FOR SYSTEMS WITH INTEGRATED WIRED AND WIRELESS CONNECTIVITY

Pablo Gutiérrez Peón

Akademisk avhandling

som för avläggande av teknologie doktorsexamen i datavetenskap
vid Inbyggda system kommer att offentligen försvaras torsdagen
den 26 september 2024, 13.14 i Pi, Mälardalens universitet, Västerås.

Fakultetsopponent: Associate Professor Federico Tramarin,
Università degli Studi di Modena e Reggio Emilia, Italy

Inbyggda system

Abstract

Ensuring sufficiently reliable and timely data communication between embedded computer systems using the wireless transmission medium has been considered problematic for decades. The open medium in which wireless networks operate makes them prone to errors, as interference from other users must be tolerated and transmission errors occur due to shadowing, fading and path loss. For this reason, key application areas such as industrial automation, automotive, avionics or robotics still rely almost exclusively on wired solutions to meet their reliability and real-time requirements. At the same time, communication technologies for consumer electronics have evolved dramatically, with a focus on higher throughput and affordable commercial off-the-shelf (COTS) hardware. This evolution is based on the success of two technologies: Ethernet and Wi-Fi. The success of Ethernet is displacing the wide range of reliable and real-time capable proprietary solutions in favor of standardized options such as time sensitive networking (TSN). With TSN, Ethernet can now meet reliability and real-time requirements, a step that its wireless counterpart, Wi-Fi, has not yet taken. This thesis addresses these limitations and opportunities and provides a comprehensive overview of how wireless communication can be introduced both as a replacement for and as a supplement to wired solutions. The thesis addresses demanding and critical scenarios where reliability and real-time guarantees are required, but where a variety of other, often conflicting, requirements may also apply. The solutions provided can be realized using COTS hardware based on Wi-Fi. They are outlined at the medium access control (MAC) level of the communication architecture, as this is important for managing access to communication resources. Several mechanisms are introduced to improve reliability and other dependability requirements by applying fault prevention and fault tolerance techniques, utilizing the diversity of communication channels, and focusing on scheduling the communication resources. The solutions are supported and evaluated by a mixture of reliability analysis, computer simulations and hardware implementation to advance the state of the art regarding dependable real-time communications for systems with integrated wired and wireless connectivity.

ISBN 978-91-7485-673-6
ISSN 1651-4238

Populärvetenskaplig sammanfattning

Datorsystem ansvarar ofta för krävande uppgifter, som exempelvis kontrollsystem i fabriker, bilar, flygplan eller i robotar. En dator kan exempelvis vara ansluten till en radar i en bil och använda avståndsinformationen för att upptäcka om det finns hinder i fordonets väg. Den kan sedan vidta lämpliga åtgärder för att undvika eventuella hinder. För att fungera korrekt måste datorn agera tillförlitligt och i rätt tid för att förhindra en olycka. Med tillförlitlig menas att ett hinder eller en reaktion inte misstolkas eller missas, och med rätt tid menas att reaktionen är punktlig så att den är klar innan en viss tidpunkt, typiskt innan bilen når fram till hindret. Det är vanligt att datorerna i olika kontrollsystem måste kommunicera med varandra för att fungera, och därmed måste också datautbytet vara tillförlitligt och punktligt. För att uppfylla de krävande kommunikationskraven i kontrollscenarier så som bilen, har man traditionellt använt sig av datornätverk som är speciellt utformade för just detta ändamål och dessa har uteslutande använt kablar. Att kablar föredras framför trådlösa lösningar beror främst på problem som störningar, som är vanligare vid trådlös kommunikation än vid överföring via kablar. Trådlösa nätverk erbjuder dock fördelar som kablar inte kan, särskilt när det kommer till mobilitet. Trots att kombinationen av trådbunden och trådlös teknik visat sig fungera väl i både hem- och kontorsmiljöer, med hjälp av tekniker som Ethernet och Wi-Fi, har inte samma sak varit fallet då olika kontrollsystem ska kommunicera med varandra. Det är värt att notera att även om Ethernet och Wi-Fi uppfattas som tillförlitliga och snabba av den genomsnittlige användaren, kan de inte uppfylla de krav på tillförlitlighet och punktlighet som kontrollsystem har. Detta beror på att utformningen av kommunikationstekniker vanligtvis inte tar hänsyn till dessa krav, eftersom de är inriktade på att skapa billiga och lättanvända tekniker för hem- och kontorsbruk.

I den här doktorsavhandlingen föreslås en uppsättning mekanismer som möjliggör att Ethernet och Wi-Fi kan kombineras och användas i kontrollsystem som kräver både tillförlitlig och punktlig datakommunikation. Mekanismerna baseras dels på att undvika att fel i form av punktlighet och tillförlitlighet uppstår, så långt det är möjligt, och dels på att hantera fel om och när de väl uppstår. I korthet använder de föreslagna mekanismerna tre huvudprinciper. För det första söker de hitta de bästa förutsättningarna för när och hur ett datautbyte ska ske, så att överföringen kan göras i tid och vara tillförlitlig. För det andra, skickas flera kopior av samma data vid olika tidpunkter så att en kopia som går förlorad på grund av störningar, kan kompenseras av

att en annan når mottagaren och systemet förblir tillförlitligt. För det tredje används en kombination av kommunikation via kablar och trådlösa nätverk för att möjliggöra datautbyte även när det trådbundna nätverket av någon anledning inte är tillförlitligt, exempelvis vid ett kabelbrott. Mekanismerna kan dessutom tillämpas även på helt vanlig elektronikutrustning som finns att köpa i handeln. För att utvärdera att de föreslagna mekanismerna verkligen är både bättre och genomförbara används en kombination av simulering och implementering på verkliga enheter. Testerna har genomförts i svåra miljöer och försvårande omständigheter, och ger lovande resultat, vilket ger vid handen att det är möjligt att genomföra krävande kontrolluppgifter med hjälp av en kombination av trådbunden och trådlös teknik.

Popular science summary

Computer systems are often responsible for demanding control scenarios such as coordinating production in factories, or systems in cars, airplanes or robots. For example, a computer might be connected to a radar sensor in a car and use the radar information to detect if there are obstacles in the vehicle's path. It may then trigger the appropriate response maneuvers to avoid them. Scenarios like these are challenging because the computer system must work reliably and timely in order to prevent an accident. Reliable means that a reaction is not missed or misinterpreted, and timely means that the reaction is completed within a certain time interval, e.g., before the vehicle reaches the obstacle. Very commonly, the computers in the different systems must communicate with each other to function, and thereby the data exchange must also be reliable and timely. In the past, meeting the demanding communication requirements of control scenarios relied on the use of computer networks that were specially designed for this purpose and operated exclusively via wires. This preference for wired over wireless deployments is mainly due to problems such as interference, which is more common in wireless than in wired transmissions. However, wireless networks offer advantages that wired networks cannot, especially regarding mobility. While the combination of wired and wireless technologies has proven itself in both home and office environments, following the introduction of technologies such as Ethernet and Wi-Fi, respectively, the same cannot be said for control scenarios. Even if Ethernet and Wi-Fi are perceived by the average user as reliable and fast technologies, they cannot meet the requirements of reliability and timeliness from control applications. This is because their design does not prioritize control requirements and instead focuses on cheaper and easier-to-use technologies for the home and office.

This thesis proposes a set of mechanisms that enable the deployment of Ethernet and Wi-Fi in combination for demanding control scenarios that require reliable and timely data exchanges. The mechanisms are based both on avoiding errors as much as possible, which is referred to as fault prevention, and on handling errors when they occur, which is referred to as fault tolerance. In short, the proposed mechanisms consist of three main ideas. First, finding the best conditions when and how a data exchange should take place so that the transmission can be reliable and timely. Second, sending multiple copies of the same data at different instances of time so that if one copy is lost to interference, another one is received and the system is still reliable. And third, using the combination of wired and wireless networks to support data exchanges when the wired network fails to provide enough reliability. Moreover, the mechanisms can be applied using commercial off-the-shelf components. To prove the usefulness and feasibility of the proposed mechanisms, a combination of analysis techniques, simulation and implementation over real devices is used. The tests are conducted in challenging scenarios and deliver promising results, indicating that it is possible to handle demanding control scenarios using a combination of wired and wireless networks.

Abstract

Ensuring sufficiently reliable and timely data communication between embedded computer systems using the wireless transmission medium has been considered problematic for decades. The open medium in which wireless networks operate makes them prone to errors, as interference from other users must be tolerated and transmission errors occur due to shadowing, fading and path loss. For this reason, key application areas such as industrial automation, automotive, avionics or robotics still rely almost exclusively on wired solutions to meet their reliability and real-time requirements. At the same time, communication technologies for consumer electronics have evolved dramatically, with a focus on higher throughput and affordable commercial off-the-shelf (COTS) hardware. This evolution is based on the success of two technologies: Ethernet and Wi-Fi. The success of Ethernet is displacing the wide range of reliable and real-time capable proprietary solutions in favor of standardized options such as time sensitive networking (TSN). With TSN, Ethernet can now meet reliability and real-time requirements, a step that its wireless counterpart, Wi-Fi, has not yet taken. This thesis addresses these limitations and opportunities and provides a comprehensive overview of how wireless communication can be introduced both as a replacement for and as a supplement to wired solutions. The thesis addresses demanding and critical scenarios where reliability and real-time guarantees are required, but where a variety of other, often conflicting, requirements may also apply. The solutions provided can be realized using COTS hardware based on Wi-Fi. They are outlined at the medium access control (MAC) level of the communication architecture, as this is important for managing access to communication resources. Several mechanisms are introduced to improve reliability and other dependability requirements by applying fault prevention and fault tolerance techniques, utilizing the diversity of communication channels, and focusing on scheduling the communication resources. The solutions are supported and evaluated by a mixture of reliability analysis, computer simulations and hardware implementation to advance the state of the art regarding dependable real-time communications for systems with integrated wired and wireless connectivity.

To my family
Para mi familia

Acknowledgments

I have dedicated ten years of work to conceive this book. I started it with a sense of entering unknown territory, with great respect for the task I was given and reasonable doubts as to whether I could complete it. Now that I have concluded it, it is difficult for me to write this without looking back at the moments and people that happened during my PhD years.

Accepting this assignment helped me explore a different environment than the one in which I was born and raised. I did not enjoy leaving what was my life behind. It took me some time to get distracted enough by the new life to stop thinking about missing out the moments with the people I had enjoyed life with until then, the people who were and still are important to me. When I left, my view on things was mostly shaped by the environment I knew – I had hardly met anyone back then who was not from there. Thanks to the PhD, I lived in two different cities, in two different countries, had the opportunity to travel and not only discover fascinating places, but also to meet people from other parts of the world, some of whom became friends. Elena Lisova, Marina Gutiérrez, Francisco Pozo and Ayhan Mehmed are the ones with whom I started the PhD and shared similar hopes and fears. I am glad that we did this together. I am thankful for the experiences that came with the PhD and for the broader and more diverse perspective on things that I have gained through them.

The years I have spent in industry (TTTech and ACDP) and at the university (MDU) have been equally rewarding. A brilliant research environment enabled me to learn the profession of researcher, make my own contributions and apply the research method in the face of the unknown as a powerful tool to address problems and shed light on them. Presenting my work at conferences is one of the experiences I will always treasure. In this research environment, I cannot help but just mention the support, guidance and patience of my supervisors Prof. Elisabeth Uhlemann (MDU), Dr. Wilfried Steiner (TTTech) and Prof. Mats Björkman (MDU). Special thanks also go to the people who conceived and managed RetNet, the project in which I started my research: Caroline Blomberg, Arjan Geven, Hans Hansson, Christian Reinisch and Carolina Reyes. A big thank you to the co-authors of my publications, with whom I really enjoyed working: Francisco Pozo, Pedro Manuel Rodríguez and Paraskevas Karachatzis. My fellow students at the university made the best out of my time in Sweden. They are an amazing group of people from all over the world who made me feel welcome from day one. Here are some names of people I got to spend more time with, a time I will always cherish. Thanks to Sara Afshar, Mohammad Ashjaei, Matthias Becker, Simin Cai, Hossein Fotouhi, Mirgita Frasheri, Svetlana Girs, Leo Hatvani,

Per Hellström, Ashalatha Kunnappilly, Nesredin Mahmud, Saad Mubeen, Apala Ray, Guillermo Rodríguez Navas, Mehrdad Saadatmand, Irfan Sljivo and Maryam Vahabi. I would also like to thank my former colleagues at University of Cantabria, especially Dr. Mario Aldea Rivas and Prof. Michael González Harbour, who helped me to take the step that has brought me to where I am today.

I would like to dedicate this PhD thesis to my family, for their love and support for as long as I can remember. I would like to thank my mum Maria José, my sister Eva, my nephews Mario and David, and my partner Dani. The hardest part of this personal journey is seeing that some people did not see it completed: In memory of my dad José and my grandmother Conchita.

The research leading to these results has received funding from the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme FP7/2007-2013/ under REA grant agreement 607727.

This work has been supported and funded by the Austrian Research Promotion Agency (FFG) via the "Austrian Competence Center for Digital Production" (CDP) under the contract number 854187.

Thank you for reading this. Peace.

Pablo Gutiérrez Peón
Napoli, Italy
May 2024

Outline

The thesis is formatted as a monograph. It is divided into two parts with a total of 10 chapters:

- **Part I. Overview.** Describes the research problem and includes a comprehensive overview of the related topics (which the experienced reader may chose to skip), with the aim of keeping a solution-agnostic perspective.
  - **Chapter 1. Introduction.** Provides a description of the research topic, the research method and a summary of the thesis contributions.
  - **Chapter 2. Fundamentals of data communications.** Provides the basics of data communications and describes in detail the subset of layers of the communication architecture that are relevant for the formulated research problem.
  - **Chapter 3. Communication requirements.** Reviews a comprehensive list of requirements placed on a communication system and concerns which help shaping the communication solutions.
  - **Chapter 4. Specialized systems: Addressing real-time and dependability requirements.** Covers the description of real-time and dependable systems due to their fundamental role in fulfilling specific communication requirements.
  - **Chapter 5. Local and personal area network technologies.** Looks at communication technologies from the local area network (LAN) and personal area network (PAN) solution domains, providing an overview of how different communication requirements are met and supplying the technology building blocks that are later used as part of the solutions.
- **Part II. Contributions.** Presents the solutions to the research problem explored in this thesis.
  - **Chapter 6. Overview of contributions.** Provides the list of the thesis contributions and a brief description of each of them.
  - **Chapter 7. Contribution area 1: detailed problem formulation, trade-offs between requirements and solutions.** Describes the research problem in depth, focusing on the requirements and the faults and failures to address, analyzes how the requirements influence each other and makes a proposal for a solution to the formulated research problem.

- **Chapter 8. Contribution area 2: fault-prevention mechanisms for the wireless medium.** Details the thesis contributions dealing with mechanisms to prevent faults when performing transmissions over the wireless medium.
- **Chapter 9. Contribution area 3: fault-tolerance mechanisms for wireless and wired networks.** Details the thesis contributions dealing with mechanisms to handle faults when performing transmissions over wireless and wired media.
- **Chapter 10. Conclusions and future work.** Reviews the research problem formulation, how the contributions have addressed it and suggests some potential future extensions.

The monograph is based on the following published papers. The list of papers is provided in chronological order:

[1] Pablo Gutiérrez Peón, Hermann Kopetz, and Wilfried Steiner. Towards a Reliable and High-Speed Wireless Complement to TTEthernet. In *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Barcelona, Spain, 2014

[2] Pablo Gutiérrez Peón, Elisabeth Uhlemann, Wilfried Steiner, and Mats Björkman. A Wireless MAC Method with Support for Heterogeneous Data Traffic. In *Proceedings of the Annual Conference of the IEEE Industrial Electronics Society (IECON)*, pages 3869–3874, Yokohama, Japan, 2015

[3] Pablo Gutiérrez Peón, Elisabeth Uhlemann, Wilfried Steiner, and Mats Björkman. Medium Access Control for Wireless Networks with Diverse Time and Safety Real-Time Requirements. In *Proceedings of the Annual Conference of the IEEE Industrial Electronics Society (IECON)*, pages 4665–4670, Florence, Italy, 2016

[4] Pablo Gutiérrez Peón, Elisabeth Uhlemann, Wilfried Steiner, and Mats Bjorkman. Applying Time Diversity for Improved Reliability in a Real-Time Heterogeneous MAC Protocol. In *Proceedings of the IEEE Vehicular Technology Conference (VTC-Spring)*, Sydney, Australia, 2017

[5] Pablo Gutiérrez Peón, Pedro Manuel Rodríguez, Zaloa Fernández, Francisco Pozo, Elisabeth Uhlemann, Iñaki Val, and Wilfried Steiner. Cognitive Radio for Improved Reliability in a Real-Time Wireless MAC Protocol based on TDMA. In *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Limassol, Cyprus, 2017

[6] Pablo Gutiérrez Peón, Wilfried Steiner, and Elisabeth Uhlemann. Network Fault Tolerance by Means of Diverse Physical Layers. In *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1697–1704, Vienna, Austria, 2020

[7] Pablo Gutiérrez Peón, Paraskevas Karachatzis, Wilfried Steiner, and Elisabeth Uhlemann. Time-Sensitive Networking's Scheduled Traffic Implementation on IEEE 802.11 COTS Devices. In *Proceedings of the International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, pages 167–175, Niigata, Japan, 2023

Contents

List of figures

List of tables

List of equations

List of algorithms

Part I

Overview

Chapter 1

Introduction

1.1 Motivation and initial problem formulation

The functioning of computer systems revolves around data, be it its processing, storage or exchange. At first, computer systems were isolated, and data exchange would have been described as communication with peripheral devices such as screens, keyboards, sensors or actuators. Nowadays, and especially thanks to advances such as the Internet, the information[1] that is available in a computer can travel much further, to where the data might be useful to another computer system and its users. Over the years, data communication has become ubiquitous and often organized in networks, with notable representatives such as local area networks (LAN) for the home and office environments, cellular broadband networks, and network deployments in factories, airplanes and cars.

Although communication networks are primarily designed for the transmission of information from senders to receivers, additional requirements may be present depending on the application scenario. Timing requirements are often considered, especially for systems that are stimulated by a constantly evolving environment and must react to these stimuli, i.e., real-time systems. An illustrative example of real-time behavior is a computer instructing a car to maneuver to avoid an obstacle. Under these circumstances, the car should initiate the maneuver before it is too late to avoid the obstacle. Is there a value in the computer to decide on the correct maneuver but not execute it in time to avoid the accident? The potentially dramatic consequences of the given scenario would lead to this system also being classified as a critical system. Requirements that fall under the umbrella of dependability are often imperative. Dependability is a property of systems that can be trusted upon to perform their function as expected. One of the most relevant attributes of dependability is reliability: What is the point of exchanging data on time if the data exchange is successful every once in a while, i.e., the system is not reliable? The term embedded systems is often used in this context when the system is limited to a very specific function, e.g., to meet real-time or dependability requirements, as in the scenario of automated car maneuvers, and in

[1] Data and information are used interchangeably in this thesis. The subtle difference between them lies in the fact that information is used to describe structured and meaningful data.

contrast to general-purpose systems.

The demand for differentiated requirements has led to the emergence of two clearly distinguishable domains: The operational technology (OT) domain, where dependability and real-time requirements are relevant and sometimes even crucial due to the criticality of the scenarios, and the information technology (IT) domain, which focuses on other requirements such as throughput and lower costs. The separation between the goals of the two domains has led them to take different paths when adopting networks to connect computer systems.

The rise of the Internet for the consumer electronics and IT sectors was timidly initiated by the emergence of modems and dial-up access to the Internet. However, the real boom came later with the introduction of the much faster Ethernet[2], standardized by the Institute of Electrical and Electronics Engineers (IEEE) in the IEEE 802.3 standard [12]. Ethernet was deployed in combination with cable modems and fiber optic networks to greatly improve the connectivity in home and office environments. These technologies have made IT the high-throughput and affordable solution based on commercial-off-the-shelf (COTS) equipment it is today, where dependability and real-time requirements are of little concern. As transmission speeds increased and the cost of consumer electronics decreased, the number of devices connected to networks continued to grow. This increase continued to a point where it became impractical to deploy a cable for each of these devices. This encouraged the introduction of wireless communication technologies, which also facilitated user mobility. Today, the success of wireless communications based on the wireless IEEE 802.11[3] standard [13] in the consumer electronics and IT sectors is undisputed.

OT, on the other hand, includes applications in industrial automation [14], aerospace [15], the automotive industry [16] and robotics [17]. What these applications have in common is the strong interaction with a physical environment where dependability and real-time requirements are needed, and the applications are often critical. Some representative technologies from the OT are PROFIBUS [18] in industrial automation or controller area network (CAN) [19] in the automotive industry. These applications have evolved significantly in recent decades as the digitalization of formerly analog functions has led to an increase in communication needs. The introduction of the Internet of things (IoT) in industrial automation, for example, aims to connect a growing number of devices with sensor and actuator capabilities. IoT is a buzzword that refers to the scenario in which any device, no matter how small, can participate in an Internet Protocol (IP) network. Another example is the use of high-speed communication backbones to connect electronic control units (ECUs) in automotive scenarios [20]. Unfortunately, OT networking has consistently been based on specialized and proprietary solutions, i.e., non-COTS, which are costly, limited in throughput and not flexible enough to cover a broader range of scenarios.

Traditionally, there has been a clear gap between the benefits of IT, which include

[2] Ethernet is described in the IEEE 802.3 standard. Although both Ethernet and IEEE 802.3 can be used to refer to the technology, Ethernet is more commonly used in this thesis as it is a widely used term in the products that support it and in academic works.

[3] Wi-Fi products are based on the IEEE 802.11 standard. In this thesis, IEEE 802.11 is the preferred term to refer to the technology as it can also help in identifying the different versions of the standard (e.g., IEEE 802.11b and IEEE 802.11g) and the predominance of the terms Wi-Fi over IEEE 802.11 is not as clear as in the case of Ethernet and IEEE 802.3.

high throughput and cheap hardware, and the benefits of OT technologies, which include meeting dependability and real-time requirements. Even though the OT and IT domains cover different application requirements and serve different application areas, a single solution that combines the advantages of both worlds is still highly desirable [21]. In this context, several attempts have been made to close this gap. These include Ethernet-based solutions such as PROFINET [18][22] for industrial environments or Avionics Full-Duplex Switched Ethernet (AFDX) [23] and Time-Triggered Ethernet (TTE) [24], both of which are mainly used in avionics. However, these technologies were developed independently from the Ethernet standardization groups of the IEEE, which later recognized the potential of Ethernet for applications outside of consumer electronics and IT. To this end, the Ethernet standard was further developed with additional mechanisms to provide time guarantees and meet reliability requirements with Time-Sensitive Networking (TSN; IEEE 802.1AS [25], IEEE 802.1Q [26] and IEEE 802.1CB [27], among others). The ongoing efforts to introduce TSN mechanisms in industrial process automation [28][29] or in the automotive industry [30][20] could be a sign that the gap between IT and OT is finally narrowing, with convergence taking place under the umbrella of IEEE standardization, the home of the successful Ethernet and IEEE 802.11.

While IT and OT finally seem to be converging, this scenario is only taking place in wired deployments, leaving wireless networks and their benefits out of the equation. Due to the characteristics of the wireless channel, packet losses are not negligible. These packet losses are the consequences of defects in the system, i.e., faults, in this case caused by the broadcast nature of the wireless channel, that increases the likelihood of transmissions being interfered. The packet losses impair dependability, which in turn has a negative impact on the fulfillment of real-time requirements. This is especially true for wireless transmissions over unlicensed spectrum, where interference from other devices is to be expected [31], but can also appear in the licensed spectrum even though transmissions by other users should be fairly limited. These determining factors are the main reason why the use of a wired transmission medium is preferred over a wireless transmission medium for embedded systems with real-time and reliability requirements. Overcoming the problems associated with the use of wireless deployments would be beneficial for extending their use beyond IT to the OT domain. However, even though wireless channels are subject to time-dependent quality changes, it is still worth noting that broken wires could permanently affect network operations.

Given the convergence of wired IT and OT communication technologies and the fact that wireless communication technologies are becoming ubiquitous in home and office environments, it is an almost natural step to address the issue of integrating wired and wireless communications for IT and OT. The goal is a solution that is able to combine the fulfillment of dependability and real-time requirements from the OT domain with the benefits of standardized and affordable high-throughput COTS devices from the IT domain. At the same time, the desired solution should not be limited to widespread wired deployments, but should also work in wireless environments to take advantage of benefits like mobility.

1.2 Research hypothesis

The research in this thesis is built on the following hypothesis:

High-throughput wireless COTS based on IEEE 802.11 can support reliable real-time communications with sufficient quality to be integrated into existing reliable real-time wired networks based on Ethernet.

The hypothesis refers to a specific class of devices on which the hypothesis can be tested. These are high-throughput wireless COTS devices based on IEEE 802.11 coming from the IT domain. The wireless devices are to be deployed together with existing reliable real-time wired networks based on Ethernet. The integration is to be achieved by enabling communications between the wireless and wired segments of the network. Communications shall observe the reliability and real-time requirements in the wireless segment, requirements that are typical to the OT domain, to a similar level of performance as the wired segment. The hypothesis therefore addresses the coverage of such requirements in unproven scenarios where high-throughput wireless COTS hardware from the IT domain is used. As a result, the solutions would integrate the benefits of wired and wireless deployments while combining the real-time and reliability requirements typical of OT with cost-effective, high-throughput standard IT solutions in one offering.

1.3 Research questions

A set of research questions (RQs) is formulated, considering the above hypothesis, to guide the research in the search for solutions to the stated problem:

- **RQ1 – On the requirements to be addressed:** *What are the trade-offs between the targeted solution requirements, in particular dependability and real-time requirements, for data communication systems with integrated wired and wireless connectivity?*

- **RQ2 – On the faults to be addressed:** *How to overcome the likely faults in systems with integrated wired and wireless connectivity so that the targeted requirements, in particular reliability and real-time requirements, are still fulfilled?* This research question is elaborated in two further questions:

  - **RQ2.1:** *What fault-prevention mechanisms can be used to support reliable and real-time communications for the wireless medium?*
  - **RQ2.2:** *What fault-tolerance mechanisms can be used to support reliable and real-time communications for systems with integrated wired and wireless connectivity?*

1.4 Research method

The research work, of which this thesis is the main result, would have been immeasurable without the backing of the selected research method, which serves as a framework providing the means to identify the problem and navigate it in the search for solutions.

Some of the reasons for such immeasurable work are the vast amount of existing technological options and scientific production in the field of data communications, which does not always allow for easy comparison, and the cumbersome process of finding a distinctive proposal that has enough weight to form a contribution.

The formulation of the research problem and the proposal of solutions followed the deductive research method [32]. In the deductive method, the hypothesis that serves as a guide for the research is formulated on the basis of existing theories. The background to these theories was obtained from a literature review on dependable and real-time communications in the LAN and personal area network (PAN) range, including wired and wireless media. The literature included scientific articles retrieved from academic works search engines after querying them with a vast list of keywords on the topic and selecting the top results. The search engine ranked the top results according to the number of citations and proximity to the keywords. Since the academic works simultaneously referenced other works, after several iterations, a comprehensive list of background publications was compiled that formed the basis for the existing theories. The result of this process was a large amount of background knowledge gained, including requirements and performance metrics that could be verified against the existing solutions. In addition, the review of related publications refined the hypothesis and helped to formulate a set of research questions. Such research questions allow the challenges to be identified, serve as assumptions shaping the problem and establish the goals that the research outcome must fulfill.

On this basis, the proposal of solutions started. Various techniques are used in this thesis to validate the solutions, including reliability analysis, computer simulations and a hardware implementation. Simulations are an excellent way to analyze the behavior and performance of the designed system. They also make it possible to detect and correct drawbacks of such a design or simply to find out to what extent the assumptions that shaped the design prove correct or not. Complementing the simulations, reliability analysis was carried out in order to obtain an approximation of the reliability performance. Nevertheless, simulations allow the exploration of a variety of complex conditions for which analyzes may not have been developed or may be too cumbersome. Unlike simulations, implementations generally do not provide the flexibility to try out different design options or replicate scenarios that would be too complex or costly, such as large networks. However, implementations are inherently more realistic, i.e., an implementation serves to uncover the challenges of an actual setup and account for the unknowns that occur when dealing with actual equipment and a real-world environment. In the simulations, for example, the precision of clock synchronization was not emulated, i.e., the clocks of all nodes were perfectly synchronized. In contrast, one of the challenges of the implementation was to synchronize the different nodes over the wireless medium with sufficient precision to follow the given schedules. Further, the simulation does not cover any other challenging aspects of the operating system or hardware. Nevertheless, simulation development can also be quite complex, especially when a high degree of similarity to reality, which can indeed be quite complex, is required. Other examples from the literature helped to find indications of the complexity and cost constraints of simulations and implementations. When the solution did not fulfill the requirements, modified solutions were proposed and evaluated until a result was obtained that was considered satisfactory.

Other works also helped to clarify the acceptance or rejection of scenarios for testing the developed simulation and implementation. If the tested scenarios did not provide interesting results, these were refined until a result was obtained that was interesting enough to show, e.g., clear differences between the scenarios.

It should be emphasized that the research presented here has not covered many of the aspects and potential solutions that the described problem would have allowed. These have been omitted both intentionally and, very likely, unintentionally. The guidance provided by the research method should help to make informed decisions about the selection and rejection of different research directions, but also shall help to minimize the impact of unintentionally missed research directions that might be relevant. Overall, the process should end with the selection and implementation of a research direction that can form a contribution, even if it is not as relevant as the initial iterations on the topic with less in-depth knowledge about it would have suggested.

Finally, there is always to a certain degree some questioning regarding the quality of the a research work like the one here presented. Among several reasons, this relates to how testable the hypothesis is and how far the reasoning behind each of the contributions is from falling into fallacies. The systematic application of a research method that has demonstrably led to high-quality results in comparable research works should reduce the reasoning flaws and make the results of different research works comparable. The research presented in this thesis also underwent extensive supervision. This supervision happened through the designated academic supervisors and the publication of peer-reviewed articles. Through this process, the author has received training as a researcher, including specialized doctoral courses on research methods and research planning. The training has strengthened the author's adherence to the research method, with all its consequences for changing the vision and understanding of science.

1.5 Summary of contributions

The thesis proposes contributions divided into three areas. Contribution area 1 (Chapter 7) explores the often conflicting requirements in data communications, in particular dependability and real-time requirements, and attempts to address them in challenging scenarios where wireless communication may be compromised by interference and other channel-related issues, and wired links may break.

The work in this thesis then proposes eleven mechanisms over high-throughput wireless COTS based on IEEE 802.11 to overcome such problems. The mechanisms are intended for the medium access control (MAC) level. The MAC level of the communication architecture is responsible for managing and scheduling the access to communication resources. The MAC level therefore plays a crucial role in meeting dependability and real-time requirements and is identified as the candidate for proposing solutions. Five of the proposed mechanisms are based on fault prevention, are published in [1][2][3][5][7] and are identified as part of Contribution area 2 (Chapter 8). Six of the proposed mechanisms are based on fault tolerance, i.e., avoiding faults when they occur, are published in [4][6] and are identified as part of Contribution area 3 (Chapter 9).

Chapter 2

Fundamentals of data communications

2.1 Data communications

Data communications[1] deals with the transmission of information units between computer systems. If computer systems lacked the means to receive input, processing could only take place on the basis of data already stored in memory. Similarly, the results of calculations would not be visible to potentially interested parties if no outputs were available. Human-machine interfaces are the means by which a person can interact with a computer system, and range from a simple button or LED light to complex virtual reality interfaces. However, humans are not the only source or recipient of data; other computer systems could also benefit from data exchanges.

Applications such as industrial process control and automation rely on multiple computer systems that are distributed throughout the production site and communicate with each other in order to be coordinated towards a common goal. For example, a production line to manufacture chocolate cakes could be designed to stop because a sensor has detected that the tank containing the cake icing is empty. Other examples include robotic systems coordinating the deployment of a communication system in an area previously covered by a mobile broadband communication system that is experiencing an outage. Transportation systems could also serve as an example. A number of trucks form a platoon and the trucks could exchange their status in order to coordinate their acceleration and braking dynamics, and save fuel by exploiting the so-called slipstream effect. In the truck itself, a series of sensors, including cameras, radars and lidars (laser-based sensors), also communicate their sensed data to a control unit, which derives status information and subsequent actions to control the vehicle. Finally, communication also takes place between PCs that are connected in an office and exchange data via email and other collaborative communication platforms. All these examples underline the intuition that networked communication systems are

[1] Most references to “data communications” in this thesis are limited to “communications” to avoid the repetition of a longer term. Further, other types of communication unrelated to data are left out of the scope.

taking on increasingly complex tasks and becoming more ordinary.

The systems mentioned in these examples are subject to a communication model that can be identified for each communication system (Figure 2.1). The model includes a source and a destination for the data. The components that send and receive data are generally referred to as nodes. The source node has a transmitter that sends the data using a transmission system via the communication channel or the link. The information is then made available to a receiver node, which in turn forwards it to the destination. Source and sender are often used interchangeably when it comes to the originator of the information. Conversely, receiver and destination are also used as synonyms for the destination of the information. It is also common to find scenarios in which the same source data is delivered to multiple destinations. The amount of data that is exchanged in the communication system in a given time interval is referred to as data traffic. The data traffic is an aggregation of data exchanges that could originate from multiple senders and could have multiple receivers. In the following sections, this communication model is enriched with the specifics of the use cases studied in this thesis.

Source
Transmitter
Channel
Receiver
Destination

Figure 2.1: Simple communication model

2.2 Communications architecture: Assigning responsibilities to layers

There is plenty of complexity involved in making data from a sender available to a receiver. For example, suppose that data is stored in the short-term memory of a computer device and the goal is for the second device to receive a copy of it. For this purpose, there could be a wired connection between the two devices. Even in a simple case without involving many of the usual communication steps such as routing, the data must at least be taken from the sender's memory and translated into electrical signals representing the data to be sent, which then propagate over the cable to the receiver. The receiving device must then be able to interpret the signals so that it attempts to recover the sent data to make it available in its memory, where it can later be checked and manipulated. A task like this must be handled by a mixture of software and hardware. Hardware will account at least for the memory, in which the data is stored, and manage the transmitter and receiver steps. General-purpose systems tend to rely more on software to solve a variety of tasks, while embedded systems tend to rely on specific hardware. An example involving general-purpose systems is when such data communication happens between two PCs, where an operating system (OS) handles several of the tasks associated with communication. Contrarily, an example of communication between two relatively simple embedded systems is when the sender (e.g., a temperature sensor) is designed to always take the data stored in the same memory area (e.g., a temperature value) and transmits it to a receiver (e.g., a thermostat), where it is always stored in the same memory area. The same process constantly takes place between these two embedded systems, and there is no need to handle it in a more generic way and with a more generic system.

Based on a minimal example such as the one just described, several sources of complexity can already be identified in data communication systems. Complexity increases with communication scenarios in which, for example, more than two devices are present, multiple transmission media are used or the applications have demanding requirements, which may include dependability and real-time requirements. This minimal example also makes it clear that the steps performed by the sender must be unwrapped on the receiving system, and that the responsibilities involved must be divided and assigned to different entities so that they can be better addressed.

After recognizing these challenges, the International Organization for Standardization (ISO) proposed a model for communications between computer systems in 1984, the Open Systems Interconnection (OSI) model [8]. The model was proposed in a context where data communication systems were still incipient and new solutions had no clear proposal that could bring a welcome consensus in a field where interoperability is greatly sought.

In such a layered architecture, each layer takes responsibility for providing a service and offers it to the upper layers while relying on the services provided by the layer immediately below (Figure 2.2). With clearly defined responsibilities and interfaces between the layers, the OSI model was intended to facilitate the development of different solutions for each layer, as they could be connected without affecting other layers. Table 2.1 provides a brief description of the individual layers. The relationship between two entities of the same layer located on different devices is governed by a protocol that "defines the rules and formats (semantic/meaning and syntactic/structure) that determine the communication behavior of the layer entities" [8]. At each layer, data from the higher layer is processed as part of the so-called payload and may also contain data required for the current layer protocol to function, known as headers, which are considered overhead. A layered architecture can also have negative effects. For example, performance could be negatively impacted as decoupled entities lack some synergies, e.g., duplicate processes and higher memory footprint.

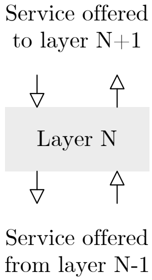


Figure 2.2: OSI layered architecture depicting the services provided and requested on each layer.

Even though the OSI model was intended to serve as a reference for the development of protocols that would eventually become the standard, this did not happen in the end. The Transmission Control Protocol (TCP)/IP architecture was already

Table 2.1: OSI layers [8]

#	Layer name	Description
7	Application	Provides all OSI services directly usable by application processes. Applications have knowledge about the entities they need to communicate with.
6	Presentation	Provides a common syntax-independent representation of the data transferred between application entities.
5	Session	Establishes a session-connection between two presentation-entities to support orderly data exchange interactions and to release the connection in an orderly manner.
4	Transport	Transfers data between session-entities and relieves them from any concern with the detailed way in which reliable and cost-effective transfer of data is achieved.
3	Network	Provides means to establish, maintain and terminate connections between multiple systems containing communicating application-entities with independence from routing and relay.
2	Data link	Provides means for the establishment, maintenance and release of data-link-connections transferring blocks of bits, i.e., messages. Detects and possibly corrects errors from the physical layer.
1	Physical	Provides the mechanical, electrical, functional and procedural means to activate, maintain and de-activate connections over the physical medium for bit transmission between data-link-entities. It is often referred to as PHY.

well established when the OSI model was conceived and provided a readily available solution for the office environment. Compared to the OSI model, TCP/IP has fewer layers, resulting in a less broad, more specific protocol set with a fusion between the layers and small shifts in responsibilities between them (Figure 2.3). For example, the application layer, the presentation layer and part of the session layer from the OSI model are merged into an application layer, which takes care of aspects such as information syntax or the establishment and release of a connection. However, the concept of layers that offer a service to a higher layer or use protocols within the same layer on different devices is also used in the TCP/IP stack. The simplified architecture offered by TCP/IP accommodates many protocols, e.g., precision time protocol (PTP), Ethernet or IEEE 802.11. However, the name only refers to TCP and IP, as these were part of the primitive offering. Even though the standardization efforts of the OSI model have not materialized in a suite of implemented compliant solutions, this should not outshine the fact that standardization organizations are behind most of the protocols used in the TCP/IP stack. Organizations such as the Internet Engineering Task Force (IETF) or the IEEE are among the most important contributors. These organizations are also responsible for creating a range of COTS devices that serve as the basis for the solution proposed in this thesis.

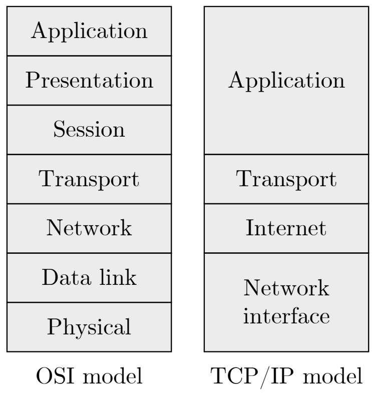


Figure 2.3: OSI [8] and TCP/IP model layers

2.2.1 Transmission medium in the physical layer

The exchange of data over a link between a sender node and a receiver node is performed across a physical path known as the transmission medium or channel (Figure 2.4). This process corresponds to the physical layer of the OSI reference model. Once the data is ready for transmission, a modulator in the sending node is responsible for generating the electromagnetic signals that represent the data. These signals, known as carriers, often have a sinusoidal shape that is repeated within the so-called carrier frequency range. The signals travel across the transmission medium to the receiving node in a process called propagation. It is important to remember that the signals are analog once realized over a physical medium, but the data they carry can have a digital interpretation. In the transmission process, the transmission time refers to the duration required to put these signals into the medium and is proportional to the amount of data. The propagation time depends on the distance between the sending and receiving nodes. For short distances and especially at high transmission rates, the propagation time is negligible compared to the transmission time. On the receiving node, a demodulator attempts to recover the data sent by the transmitter. Although the data is represented in binary format, the realization of the signals in the physical world is part of a continuum. Nevertheless, digital communications deal with a finite set of combinations for the transmitted data. Consequently, the receiver is not expected to discern between an infinite number of possibilities, but from the finite set of possible messages that the sender could have transmitted. The various steps associated with the physical layer have traditionally been covered in hardware, but the increasingly common SDRs allow some of these aspects to be handled in software.

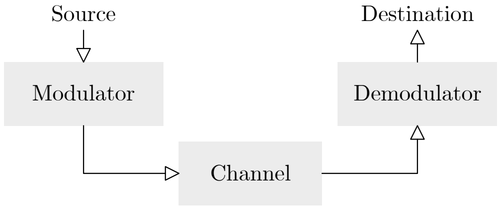


Figure 2.4: Simplified block diagram of a digital communication system

Noise in communications

The transmitted analog signals always contain noise (Figure 2.5). There is a particular type of noise, known as thermal noise, which cannot be eliminated and is referred to as additive white Gaussian noise (AWGN). The signal-to-noise ratio (SNR) measures the level of the signal in relation to the level of the background noise. The SNR correlates with the number of erroneous bits and is used to measure the quality of the transmission medium. However, SNR values are not static, but are subject to dynamic changes due to fluctuations in the noise level in the time, space and frequency dimensions. There are several reasons for low SNR values, including low transmission power, path loss, shadowing, multipath fading and interference [33] (Figure 2.6). Low SNR values are therefore problematic, while unstable SNR values are detrimental to determinism.

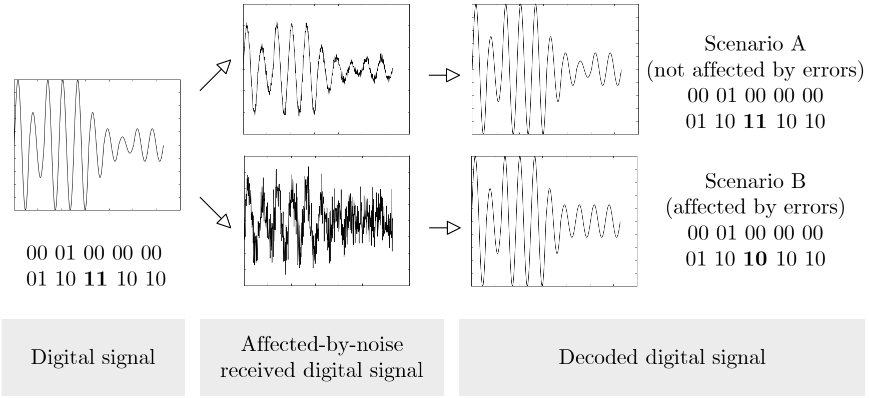


Figure 2.5: Decoding a digital signal affected by noise. If the noise is severe enough, the decoded signal may contain errors.

The transmission of a low power signal is common in energy-constrained environments where low throughput and short ranges are assumed. Path loss refers to the decrease in signal strength as it moves away from the transmitter, i.e., it depends on the distance between the transmitter and receiver. It is particularly noticeable for unguided media, i.e., wireless media, where the signal often propagates in multiple directions, dissipating more power. Shadowing and multipath fading are also problematic for unguided media. Shadowing is caused by physical obstacles between sender

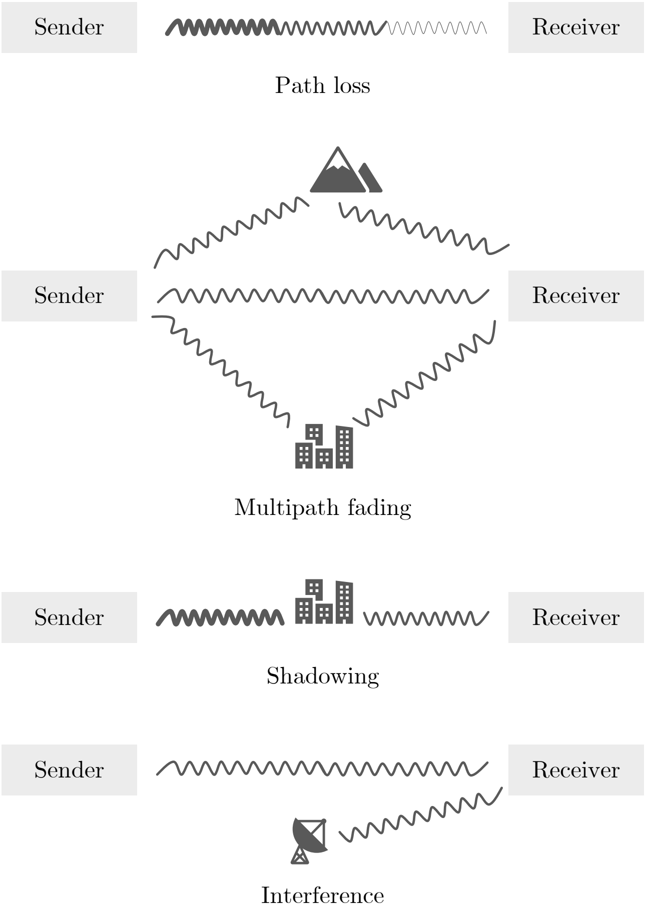


Figure 2.6: Representation of signal path loss, multipath fading, shadowing and interference

and receiver that reduce the strength of the signal, redirect it or block it entirely from reaching the receiver. Multipath fading results from the signal taking different paths towards the receiver, paths of different lengths, which consequently take different times. The receiver perceives all these signals as overlapping, which may have a positive or negative effect on the process of distinguishing the initially transmitted signal. While multipath fading and shadowing can be a relevant problem, interference caused by competing signals in overlapping frequency bands is a major issue for transmission media shared by several transmitters, i.e., unguided media and bus topologies, where the transmitters are part of a broadcast area. Interference can distort or cancel out a signal. Interference can originate from nodes in the communication system that transmit simultaneously due to a lack of coordination or a coordination mechanism that does not fully prevent simultaneous transmissions. Another source of interference that only affects unguided media is simultaneous transmissions on the same frequency from devices that are not part of the communication system and use the same or even a different communication technology. In such scenarios, the coordination mechanisms can only rely on a reactive approach based on the current state of the medium. For example, it is common to sense the state of the medium for a certain duration and transmit if no other transmissions have been detected. If other transmissions are detected, the transmission is attempted at a later time. Interference can be particularly damaging in industrial environments where there is a harsh environment full of machinery and electrical components [14]. Interference can also be significant in the industrial, scientific and medical (ISM) radio bands of the unlicensed spectrum used by technologies based on the IEEE 802.11 and IEEE 802.15.4 standards, such as Wi-Fi and Bluetooth respectively, where access to the medium is free and an unlimited number of devices can attempt to transmit simultaneously [31]. Therefore, sometimes it cannot be guaranteed that the environment is free from interference.

The wired and wireless channels can be modeled by considering the effects of AWGN. However, due to the more prominent disturbance on the wireless channel, other models are often required to better represent specific wireless channels.

Improving SNR and its positive impact on transmission reliability is crucial to ensure that the communication system can meet the requirements of dependable and real-time applications. At the physical-layer level, it is common to have two additional steps before modulation and after demodulation, the encoding and decoding processes, where redundant information is used to protect the transmission from channel problems. Channel diversity-based mechanisms are also a powerful tool to increase the probability of successful transmissions.

Transmission diversity

The SNR of the channel is subject to variations depending on the instant when the signal is transmitted, the frequency of its carrier and the path it uses to propagate. These variations range from barely noticeable to so pronounced that they could impair the transmission. Thus, a communication system often needs to decide the best combination of time, frequency and space resources usage to carry out transmissions. Transmission diversity schemes (Figure 2.7) attempt to exploit such variations to achieve better SNR by covering a larger number of channel conditions. A diversity scheme can increase the probability of successful delivery for a single transmission by selecting the time, frequency and spatial path that maximizes the probability of success. Alternatively, diversity is often used to send multiple copies of the same data subject to different channel conditions so that the data exchange is less exposed to temporal or permanent problems affecting a particular frequency or propagation path.

$$\text{Transmission diversity schemes}\begin{cases}\text{Time diversity}\\ \text{Frequency diversity}\\ \text{Space diversity}\end{cases}$$

Figure 2.7: Transmission diversity schemes

Although sending additional copies increases the probability of successful delivery, communication systems often set a limit on the number of copies to avoid endless transmission attempts. This limit is particularly important if a deadline applies to the transmission. In this situation, the transmission is considered to have failed if the maximum number of transmission attempts is reached or the deadline is exceeded. In addition, the copies could be sent unconditionally if the sender receives no feedback on the success of the transmissions. In systems with such feedback, the sender could stop sending further copies after a successful delivery. Even if these feedback mechanisms cause a certain overhead, this is often compensated by the throughput that is freed up when saving unnecessary copies of the same transmission.

Transmitting copies of the same data at different instants, i.e., time diversity, is a common method to increase the probability of successful delivery, even if this has a negative impact on delivery latency, delivery jitter and throughput. The effects on delivery latency and jitter are particularly relevant for systems that have to observe time constraints. For these systems in particular, the decision on the time at which the multiple copies are sent is of crucial importance. The decision may not only be based on the timing requirements, but may also reflect on the channel conditions. For instance, interference usually occurs in sudden bursts for a particular duration. Thus, consecutive transmissions may not be the best approach, as the burst that interferes with one transmission is more likely to interfere with subsequent copies. In such cases, it can be useful to separate the different copies in time.

With frequency diversity, the communication system uses different carrier frequencies to transmit data. In general, an antenna can only operate on a single frequency at a time, so transmissions at different frequencies have to be carried out at different instants. Non-overlapping frequencies can be used in parallel, but this approach requires the existence of pairs of send and receive antennas for each of the frequencies used in parallel. Frequency-hopping spread spectrum (FHSS) is an example of a widely used mechanism to protect transmissions from interference by rapidly changing the carrier frequency following a defined sequence. Another mechanism to improve reliability is cognitive radio (CR). Compared to FHSS, cognitive radio does not transmit on all available frequencies, but tries to find the one that maximizes the probability of a successful delivery. Cognitive radio is based on dynamic spectrum management, in which the frequency is selected dynamically depending on the regularly retrieved information on channel usage. Once this channel status information is retrieved and processed, a decision is made on the frequency that provides the best channel quality and transmissions are made based on this decision. Cognitive radio systems were initially conceived as an approach that takes advantage of licensed channels where external transmissions are not allowed to avoid interference with primary users. However, they were later proposed to improve reliability by avoiding interference in ISM bands [34]. With the cognitive radio approach, the external transmissions can still take place as long as the primary user is not interfered.

In space diversity schemes, data is transmitted via different physical paths. A prominent example is the so-called multiple-input and multiple-output (MIMO) systems, in which nodes with several wireless transmitters and receivers convey data using different physical paths. Similar to the use of multiple frequencies, the physical paths can be used to either increase overall throughput or improve reliability by sending redundant information without introducing further delay.

Collision domains

The channel is shared by all nodes that belong to the same collision domain. A collision domain refers to the combination of time, frequency and space dimensions in which transmissions cannot take place simultaneously because, as the name suggests, they would collide. To prevent the transmissions from colliding, they shall not coincide in their time, frequency and space dimensions (Figure 2.8). The fact that several nodes belong to the same collision domain is an advantage when several receivers expect the same data transmission, e.g., in broadcast transmissions. However, it becomes a limitation if the recipients are to receive different data, e.g., for unicast transmissions. For any pair of nodes that can communicate directly over a transmission medium, communication is said to be full-duplex if both systems can send and receive data simultaneously. In half-duplex systems, the two nodes must swap the roles of sender and receiver, as the transmissions cannot take place simultaneously. Half-duplex communication is typical of wireless media, where a single antenna is usually unable to send and receive data simultaneously, and transmissions often use the same frequency and are therefore part of the same collision domain. Nevertheless, it is important that all wireless nodes that want to exchange data are installed in such a way that they are in an area where they can reach each other. If the wireless end systems belong to groups that do not communicate with each other but are in the same area, the transmissions must use different frequencies or coordinate the transmissions to avoid being part of the same collision domain.

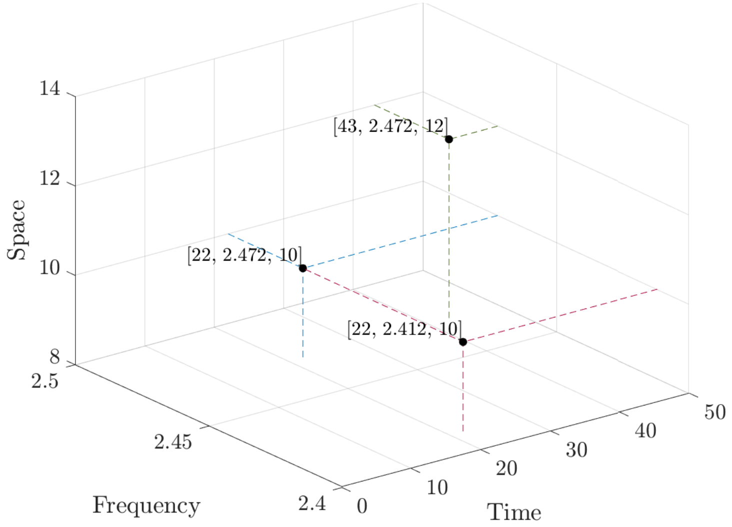


Figure 2.8: Visual example of transmissions happening in different collision domains. To avoid overlaps, the combination of time, frequency and space dimensions for different transmissions cannot be the same.

Wired vs. wireless channel

Some of the characteristics of wired and wireless channels have already been described until now and give an idea of the significant differences between them. Table 2.2 gives an overview of the comparison between them.

In general, wired communications is less prone to fluctuations in SNR and is therefore considered more stable. A wired medium is also considered an advantage since it allows establishing links between nodes that exchange data in parallel to other wired links without interfering with each other. Such connections often have a higher speed than wireless connections. The full-duplex wired connection allows the easy establishment of multi-hop topologies that can cover a larger number of nodes and distances, provided that the intermediate nodes and links are capable of handling data converging from several sources and potentially having multiple destinations. These reasons, and especially SNR stability, make wired networks the preferred option for systems with dependability and real-time requirements. However, a broken cable can be expected to disrupt communication between two ends unless an alternative path is available for the data, which is an important limitation of the wired channel.

Wireless deployments come with benefits not easily found in wired settings, such as mobility, more flexible deployments, lower installation costs, easier maintenance or lack of problems due to broken wires. However, they generally have poorer SNR values caused by a more prominent path loss, multipath fading and shadowing. The more volatile SNR values are often seen as an obstacle for applications with dependable and real-time requirements. Degraded reliability due to interference is a problem that is exacerbated in frequency bands that are not protected but have an open access policy, such as the ISM bands. Depending on the extension of the wireless communication system, the interference problem can be limited to a specific area, but it may also jeopardize a multi-hop data exchange if one of the links involved is affected by the interference. It has also been shown that neighboring channels in technologies such as IEEE 802.11 and IEEE 802.15.4 also interfere [35] [36]. The broadcast nature of wireless communication systems makes the connections to be half-duplex, which results in an important limitation. The varying SNR values also affect the transfer rates, which are often subject to changes to adapt to channel conditions, making the already lower transfer rates even lower compared to wired settings. Range is also time-, space- and frequency-dependent, so nodes at the limits of range are constantly switching between reachable and out-of-range. Wireless communication systems are also subject to security concerns due to the unguided media, with problems like eavesdropping or jamming demanding protective solutions. Given the many obstacles that can arise when deploying a wireless communication system, it is critical to characterize the behavior of the system in each scenario and anticipate countermeasures. It is particularly important to apply mechanisms to improve reliability when the communication system must meet dependability and real-time requirements.

2.2.2 Medium access control in the data-link layer

Directly above the physical layer, the data-link layer is responsible for establishing data connections, i.e., links, between communication devices. The layer manages access to the link by selecting the moment when the transmission occurs. Having a protocol

Table 2.2: Comparison between wired and wireless channels. The "+" symbol means that the aspect generally performs better in the respective channel, while the "-" symbol stands for a generally worse outcome.

Aspect	**Wired channel**	**Wireless channel**
Broadcast transmissions	-	+
Deployment flexibility	-	+
Duplex	+	-
Installation cost	-	+
Interference	+	-
Maintenance easiness	-	+
Mobility	-	+
Multipath fading	+	-
Path loss	+	-
Permanent channel faults	-	+
Range	+	+
Security	+	-
Shadowing	+	-
SNR levels	+	-
SNR stability	+	-
Throughput	+	-
Unicast transmissions	+	-

ruling the access to the medium is especially relevant when the medium is shared by multiple participants, such as in a bus topology or with unguided media. In the terminology of the widely used IEEE 802 standards, the responsibility for the access to the medium is assumed by the MAC layer as part of the link layer.

The MAC layer is crucial to fulfilling several of the communication requirements. Coordinating the medium access is essential to ensure that data is delivered with a specific latency and jitter or before a delivery deadline, after considering aspects like their data occurrence pattern or criticality. Such coordination is frequently backed by mechanisms based on admission control and scheduling algorithms. Admission control manages the amount of data transmitted by the nodes. This is helpful to not exceed the available channel bandwidth and can also help to decide what application shall consume the bandwidth by following different criteria, e.g., admitting traffic based on criticality or fairness. Scheduling algorithms allocate the use of communication resources to the different data exchanges. For example, the scheduler could be responsible for selecting which data exchange is using a link at every time or what location in the memory, i.e., in the queue, it shall store it.

The MAC layer often has to find workarounds for the limitations coming from the channel or the architecture of the nodes. For example, a limitation in wireless settings is the hidden node problem, where an intermediate node can communicate with other nodes. However, the latter cannot communicate with each other, which could result in them trying to transmit to the intermediate node at the same time and interfering with each other in the meeting point. There are solutions to this problem, such as the request to send (RTS) clear to send (CTS) of the IEEE 802.11 standard. Another example of a limitation is the existence of single queues for handling all data to be transmitted, which becomes problematic if data with different time and criticality requirements is to be processed (Figure 2.9). A situation could arise where high-criticality traffic is periodically forwarded to a node, but it has to share a single queue with low-criticality traffic that occurs in an unbounded fashion. Such behavior might compromise high-critical traffic to the point that it has to be discarded because the queue has a limited size and could be filled by the low-critical traffic. This situation could be solved by splitting the traffic into two or more queues, each serving a different criticality.

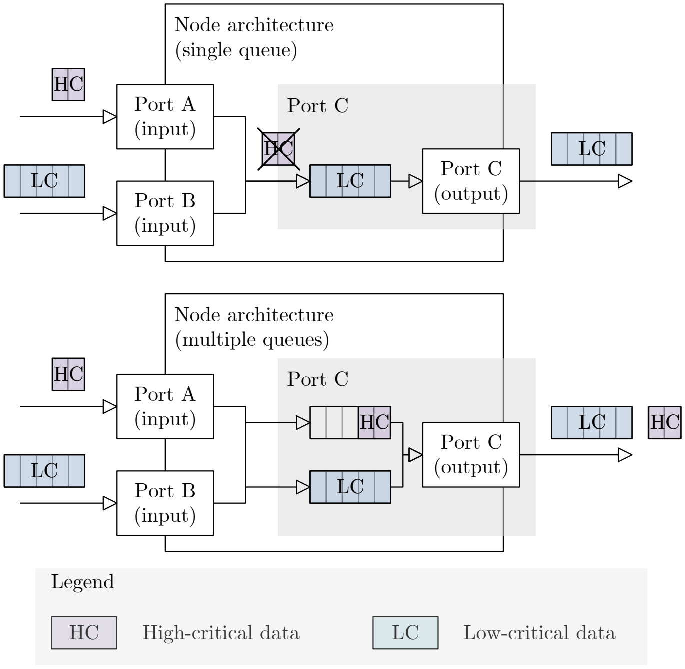


Figure 2.9: Scenarios with one or multiple transmission queues in the node architecture

Coordinating the medium access is likely to be beneficial for reliability, availability and determinism, e.g., by preventing transmissions coming from multiple nodes from overlapping. Nevertheless, even if access to the medium is granted by the MAC protocol, poor channel quality can jeopardize the transmission, e.g., due to interference caused by collocated communication devices. Therefore, reliability improvement mechanisms based on the fact SNR varies depending on the time, frequency or location are often used at the MAC layer.

Taxonomy

A common way to classify MAC protocols (Figure 2.10) is by assessing how they handle contention when two or more nodes have data ready to transmit within the same time interval. Contention-based protocols do not guarantee that the medium is free from collisions caused by other nodes in the communication system during a transmission. Contention-free MAC protocols, on the other hand, offer such a guarantee. Overall, the MAC protocol cannot always completely avoid collisions with devices outside the control communication system, but it can be designed to cope better, e.g., by allowing redundant transmissions of the same data.

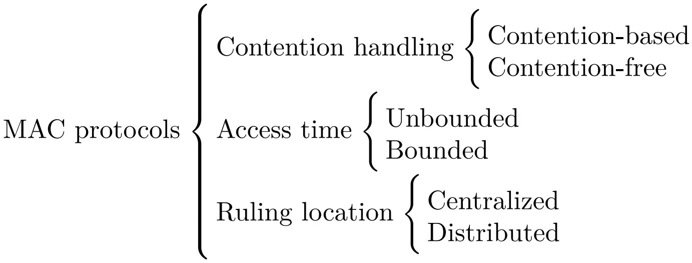


Figure 2.10: A taxonomy of MAC protocols

An orthogonal classification for the MAC protocols concerns the time required to access the transmission medium. According to this classification, a distinction is made between protocols with unbounded or bounded access time. MAC protocols with unbounded access time follow a reactive approach in which the transmission depends on whether other nodes are transmitting at that time. As the other nodes could have data to transmit at any time, a transmission attempt might be delayed indefinitely by transmissions from other nodes. In MAC protocols procuring bounded access time, the access time of a node is not dependent on other nodes. The bounded access time guarantee only applies if the amount of traffic sent by the nodes does not exceed the capability of the links and nodes to handle the data. The resources are then divided between the transmissions so that the transmissions do not interfere with each other.

A third perspective on the classification of MAC protocols focuses on the way the protocol ruling is conducted across nodes, distinguishing between centralized and distributed protocols. In centralized protocols, the actions that govern the MAC are decided by central nodes that arbitrate the access to the medium for the rest of the nodes. For example, the central node can send notifications to other nodes to trigger their transmissions. In contrast, with distributed protocols there is no hierarchy for the nodes, but they all coordinate based on a set of rules that can be followed independently. An example of a distributed MAC protocol is when the nodes only transmit after they have sensed the channel and have not detected any interference.

A MAC serving real-time applications should opt for a contention-free approach with bounded-access time. Such an option makes it possible to calculate a worst-case time for accessing the medium. In addition, different MAC strategies can be applied to different types of traffic and used together to provide a MAC protocol that supports applications with different reliability and real-time requirements.

Baseline protocols

ALOHA was the first attempt in the early 1970s to establish a wireless data communications network connecting users in the Hawaiian islands. In its first version, the ALOHA MAC protocol was based on sending when a node had something to transmit. The sending node senses the medium during transmission to check for possible collisions. If there are collisions, the node attempts to transmit later. A version called slotted ALOHA modified the first protocol version to reduce the likelihood of collisions. In slotted ALOHA, transmissions only begin at predefined times, creating temporal slots that reduce the probabilities of overlapping transmissions and increase overall throughput. In addition, slotted ALOHA requires the nodes to be synchronized so that the time slots between the nodes are aligned.

Inspired by the development of ALOHA, the carrier sense multiple access (CSMA) protocol was incorporated to the family of contention-based random access protocols that are widely used in current non-real-time setups. In CSMA protocols, the channel is first sensed for some time before attempting to transmit to determine whether other users are transmitting. This procedure avoids a problem in ALOHA networks, namely starting a transmission when another transmission is in progress that could cause a collision. CSMA is usually offered in two variants: collision detection (CD) and collision avoidance (CA). CD is obviously better suited to avoid collisions but requires sensing the channel while transmitting, a feature that is not yet widely supported in wireless communications. In the case of CSMA/CA, the medium shall be free for the duration of the so-called interframe space (IFS) T_{IFS} before transmitting. If the medium is sensed as busy during T_{IFS}, the node backs off, i.e., desists from transmitting, for a random time T_{CW} taken from the range given by the so-called contention window (CW) CW, which is defined between the limit values provided by CW_{min} and CW_{max}. More precisely, T_{CW} takes a random value from a CW between 0 and CW_{min} the first time the node backs off. The CW is extended on every failed transmission attempt up to a limit given by CW_{max} and brought back to the initial value between 0 and CW_{min} if a transmission is successful. CSMA/CA is a straightforward mechanism for accessing the medium that serves its purpose in situations without congestion. However, the backoff mechanism can make the transmissions very complicated in situations where multiple nodes are trying to access the medium. Furthermore, this mechanism is not suitable for real-time traffic, as access to the medium is unbounded.

Token-passing protocols are one of the alternatives that enable real-time traffic. These protocols use a token that circulates between the nodes and follows a predefined route, often based on the simple and fair round-robin assignment. Only the token holder at the time can transmit, effectively avoiding collisions. However, the circulation of the token causes significant delays and is subject to considerable jitter. In addition, an interfered token might cause the network to lose track of which node is the next holder, which would require a mechanism to recover the token.

Polling-based protocols could also provide the determinism that real-time traffic requires. A node operating with a polling-based MAC protocol transmits only after receiving a triggering message, i.e., a polling message. A single coordinator is generally in charge of sending the polling messages. In practice, collisions can be avoided if only one node receives the polling message at a time. Unfortunately, the protocol introduces considerable overhead as a polling message is required to trigger data transmissions.

Slotted ALOHA, token passing and polling-based MAC protocols can be considered as a subset of time-division multiple access (TDMA) protocols. TDMA is based on the multiplexing the access of different nodes by dividing the available time into time slots. A deterministic allocation of time slots is crucial for avoiding conflicts and providing real-time guarantees. A mechanism based on token passing or polling messages can be used to enforce the allocation of time slots. An alternative that significantly reduces overhead is to allocate time slots based on a sequence known to all nodes, e.g., round-robin, which does not require or drastically reduces the exchange of protocol messages during operation. If the assignment of slots is complex enough, the complete schedule can also be pre-installed in the nodes before the start of operation. Sometimes the schedule needs to be changed after some time to adapt dynamically to changes in the requirements. In that situation, so the protocol needs to update the schedule by sending it to the nodes. A common requirement of TDMA protocols serving strict timing applications is to keep the same time notion between the nodes so that the time slots between the network nodes are aligned. In the simplest variant, a message sent by a central node, e.g., the polling message, can synchronize the rest of the nodes. However, more complex synchronization protocols can also be used to support cases like multi-hop networks, but would entail some overhead.

Table 2.3 summarizes the main characteristics of the baseline MAC protocols and families.

Table 2.3: Baseline MAC protocol families and their main characteristics

Protocol / protocol family	**Principle**	**Contention handling**	**Medium access time**	**Ruling location**
ALOHA	Send as soon as having data to transmit.	Contention-based	Bounded	Distributed
Slotted ALOHA	Send as soon as having data to transmit but only start in predefined slots.	Contention-based	Bounded	Distributed
CSMA	Send as soon as having data to transmit and no other node is transmitting.	Contention-based	Unbounded	Distributed
Token passing	Send as soon as having data to transmit and having the token.	Contention-free	Bounded	Distributed
Polling-based	Send as soon as having data to transmit and having received a polling message.	Contention-free	Any	Centralized
TDMA	Send inside time slots.	Any	Any	Any

In addition to the improvement in reliability that comes with a contention-free protocol, MAC protocols often rely on retransmission mechanisms to further improve reliability. For example, the automatic repeat request (ARQ) is a mechanism that retransmits data for which the sender has not received an acknowledgment (ACK).

Several examples of MAC mechanisms can be found as part of the review of different technologies in Chapter 5 and in the related work sections in the contributions part (Part II).

2.2.3 Network layer

Computer systems that exchange data with each other are usually organized in communication networks. Networks are outlined directly above the data link layer of the OSI protocol architecture. These interconnected systems, commonly referred to as nodes, are further classified into two categories. The nodes that serve as data sources or destinations are commonly referred to as end systems, endpoints, or talkers and listeners. The terminology often depends on the author's preference and the conventions in the respective field and communication technology. On the other hand, there are often intermediate nodes that forward the data on its way from the source to the destination. These nodes are referred to as switches, routers, gateways or repeaters. In this case, these terms have a strong connotation that refers to the specific function they perform. Switches and routers forward data according to the logic defined at the data-link and network layer of the protocol architecture, respectively. In contrast to switches and routers, repeaters forward data on all possible paths without taking the actual destination into account. Gateways, on the other hand, are used to connect two different network technologies, e.g., Ethernet and IEEE 802.11.

There are multiple options, i.e., topologies, to connect several end nodes that want to exchange data (Figure 2.11). In fully connected topologies, links are arranged between each pair of end systems providing a dedicated connection. In wired settings, this translates to having a cable between each pair of nodes, a solution that does not scale with the number of nodes. In wireless settings, the use of point-to-point connections benefits from the broadcast nature of the wireless channel. However, wireless point-to-point transmissions are limited by the bandwidth of the channel, which is shared by all nodes, and by the range.

The use of intermediate nodes enables the deployment of more flexible topologies and more extensive networks. Intermediate nodes enable simultaneous transmissions on different links, but at the cost of increasing the number of transmissions required to reach a destination and thus also the latency. Scalability is one of the main reasons for the popularity of star and mesh topologies over other previously common topologies such as bus, daisy chain, ring or tree. In star topologies, the end systems are connected to an intermediate node. Star topologies can also replicate the connection pattern to have several intermediate nodes, creating a start of stars. Mesh topologies are often found in wireless deployments, where the end systems also act as intermediate nodes and can forward data to the nearby nodes. In contrast, bus topologies connect all end nodes with a single cable without intermediates. Although a bus topology is simple, it does not scale well when the number of nodes increases as it cannot perform simultaneous transmissions. In contrast, the daisy chain connects the nodes linearly like a bus, but requires each node in the chain to pass the data to the next until the destination is reached, allowing a certain level of parallelization but incurring in relaying overhead. Despite the often large number of hops in star and mesh topologies, their throughput is not affected as much as in a bus topology or daisy chain when the

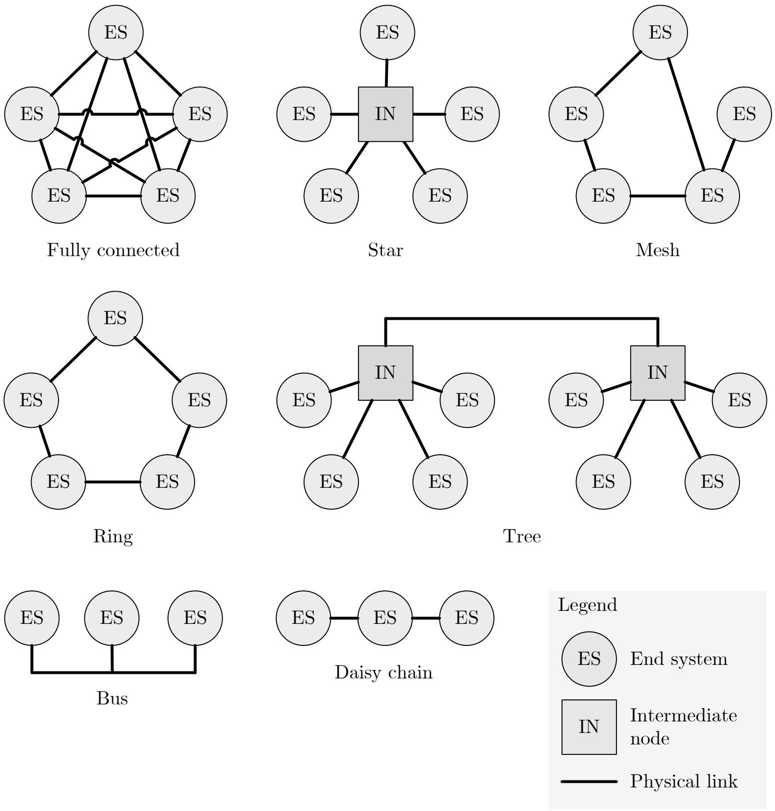


Figure 2.11: Network topology types

number of end systems increases. Ring topologies establish two links for each node, with the end systems acting as intermediate nodes. Each node in a ring can be reached from either of the two connected links, which resembles a daisy chain with the ends connected. Like bus topologies, ring topologies do not scale well because the average number of nodes a transmission shall go through increases linearly with the number of nodes. Tree topologies connect their nodes as in a star network, whereby intermediate nodes can be connected in between. However, tree topologies traditionally represent the combination of several star-shaped networks connected via a bus, with the end systems assuming the role of intermediate nodes, inheriting the limitations of the bus topology. In wireless settings, a different terminology might be used to describe the topologies. For example, IEEE 802.11 speaks of an ad-hoc topology when two end systems communicate directly with each other without intermediate nodes. IEEE 802.11 also describes the infrastructure topology, in which end systems communicate via an intermediate node, thus defining a star topology. The topologies mentioned can also be combined, e.g., interconnected wired and wireless topologies.

Arranging a topology usually requires some configuration, e.g., running some routing protocols to learn the routes between the different nodes or having a mechanism to associate wireless nodes together and dynamically build the network. There are often two options for such a configuration, considering the moment in which it occurs: either a static configuration before the network is started or a dynamic configuration after the network has been put into operation. In the first case, the network can start exchanging data immediately after start-up. With the second option, it may take some time before the configuration is applied and the network is ready for operation. The dynamic configuration has the advantage that it can react to changes in the network, e.g., the addition and removal of nodes. Unfortunately, it can be difficult to obtain some of the required configurations at runtime, e.g., a network configuration including scheduling to fulfill dependability and real-time requirements. The concept of software-defined networking (SDN) is often used to refer to network technologies that are dynamically configured.

The technology used in a network deployment differs dramatically depending on the size of the physical area that the deployment needs to cover. In the case of embedded systems, the deployment is physically constrained to the system to which the embedded computer belongs, e.g., a vehicle, a robot or a building. Hence, some embedded systems are served via LANs. LANs provide connectivity to computer systems within a single area, namely a residential unit, a business or an institution. LANs range in size from a few square meters to a large building or series of buildings, connecting tens to thousands of devices. In contrast, PANs are deployed in smaller areas, usually a few square meters, often serve embedded systems and are rather limited in terms of throughput or energy consumption. In the case of wide area networks (WAN), they cover large geographical areas and are used to connect individual computer systems or LANs at a greater distance. Differences in aspects such as signal power, energy consumption or hardware are the reason why PANs, LANs and WANs use different network technologies. A detailed description of selected technologies from the LAN and PAN area can be found later in Chapter 5.

Until now the term link has been used to refer to the physical connections between devices, either via cable or a wireless channel. However, another dimension of topologies is often considered and is worth mentioning. The logical topology refers to the connections between nodes as seen by the upper layers in the communication stack. Virtual LANs (VLANs), defined by a number of protocols including IEEE 802.1Q, are often used to create such topologies for resource partitioning and security reasons.

Given a set of end systems, the problem of connecting them optimally is extremely complex given the immeasurable number of possible combinations. The potential network deployments could include combining different topologies, offering diverse physical-link layers or providing redundant paths to reach a destination as a mechanism for increased reliability. The selection of communication technologies based on the communication requirements could help to narrow down the problem by restricting the topologies to the alternatives offered by these technologies, e.g., switched Ethernet. However, after limiting the number of options, it is likely that the remaining number is still too large to evaluate the suitability of each of them. In this context, it is common to get inspiration from deployments covering similar requirements and that have already proved to operate correctly. For example, in automotive networks, it is common to take a zonal approach where intermediate nodes are placed based on the different areas in the vehicle where connectivity is required, e.g., front and rear networks. Another common approach in automotive networks is when intermediate nodes are placed to connect end systems that share a common functionality, e.g., powertrain functionality and perception of the surroundings functionality. Different communication technologies might also serve as a partition criterion. For example, by putting together nodes that use the same communication technology like CAN or Ethernet. In this context, a gateway takes over the translation between the different networks. Ultimately, the performance of these deployments needs to be analyzed, tested or simulated in realistic scenarios to prove their validity.

When exchanging data over the network, a distinction is made between the addressees of the data. The addressing scheme is referred to as unicast if the data has a single destination, as multicast if it has multiple destinations, and as broadcast if all possible recipients are addressed.

When multiple end systems send and receive data over the network, the links and intermediate nodes are likely to be shared by multiple transmissions. The shared paths that are established between each pair of communicating nodes require multiplexing, a mechanism by which multiple transmissions can utilize the shared communication resources. Multiplexing manages in the network layer the time, frequency and space dimensions of the transmission channel that are to be used by several transmissions. Different methods can be used to select the time, frequency and physical path for a transmission. The earliest networks were based on circuit switching, where a path for exclusive use between the sender and receiver nodes for the duration of the communication was established. Circuit switching has the major disadvantage that the segments in the path are not available for transmissions from other nodes during the time they are reserved. In contrast, the packet switching method allows the paths to be shared by handling chunks of data, i.e., packets, that have a destination address used to redirect them through the network. Each packet requires a combination of time, frequency and physical path that must be different from those of other packets to avoid collisions. In this way, packet switching enables the simultaneous transmission of packets having different sources and destinations. However, packet switching networks are still limited by the throughput provided by each of the links and the processing and storage capacities of intermediate devices, which may not be sufficient to handle all packets during periods of high load. Therefore, scheduling the utilization of communication resources in networks where multiple nodes exchange data packets is crucial to ensure meeting dependability and real-time requirements.

2.2.4 Handling of communications in software: the role of operating systems

Operating systems are often involved in data communications (Figure 2.12). They are available for general-purpose and some embedded systems, and offer services that applications can benefit from including communication management. A common abstraction used by operating systems for communications is network sockets. They play a central role in computer networks as they abstract applications that want to send and receive messages from the details of network protocols and hardware implementations, and arbitrate the access to the network resources. Additionally, in operating systems such as Linux, the software gap between sockets and hardware is not left to a regular program, but is covered by the operating system itself. The main reason for the operating system taking responsibility is that network interfaces are shared and deal with messages coming and leaving asynchronously that involve different users. Operating systems also provide some additional services that all socket applications benefit from, like address resolution and routing.

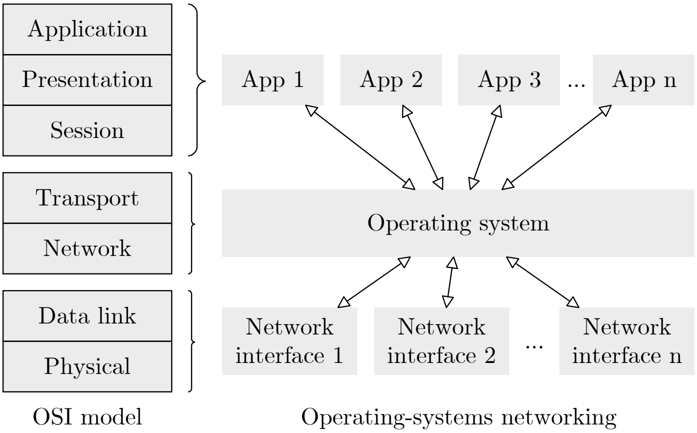


Figure 2.12: Operating systems networking

Chapter 3

Communication requirements

The characterization and selection of the requirements that a data communication system shall fulfill is crucial to nominate the communication technologies providing a fitting solution. Some of the requirements might be relaxed and selected as optimizations that are not essential to the functioning of the system, but are considered nice-to-have or candidates for future improvements. At the same time, the performance of the system can be measured based on the degree of fulfillment of the requirements and optimizations.

Although the requirements can be very specific to the use case and are therefore numerous and diverse, the ones listed below are selected due to their relevance for dependable and real-time communications. Figure 3.1 shows the communication requirements selected for discussion, classified according to the typical split between functional requirements, which define what the system shall do, and non-functional requirements, which relate to how it is done. Non-functional requirements do not refer to the basic functionality of the system, but to the quality of the solution or to constraints related to, e.g., resource considerations or technical limitations. Figure 3.1 then groups the requirements according to their semantic similarity into data characterization, data delivery, dependability and others.

Next, each of the requirements is described in detail. For each requirement, a table is used to summarize key aspects. These aspects include related terms that are usually found in relation to the requirement in question, whether the requirement is labeled as functional or non-functional, whether the requirement can be selected as an optimization, if the requirement can be used as a performance metric to measure the performance of the system, and finally how the requirement can be characterized. After describing the requirements, the last part of the chapter introduces the concept of traffic classes as one of the terms that communication technologies use to specify their offering to fulfill the requirements.

3.1 Data characterization requirements

The requirements under data characterization describe the properties of the data that are relevant for proper handling by the communication system.

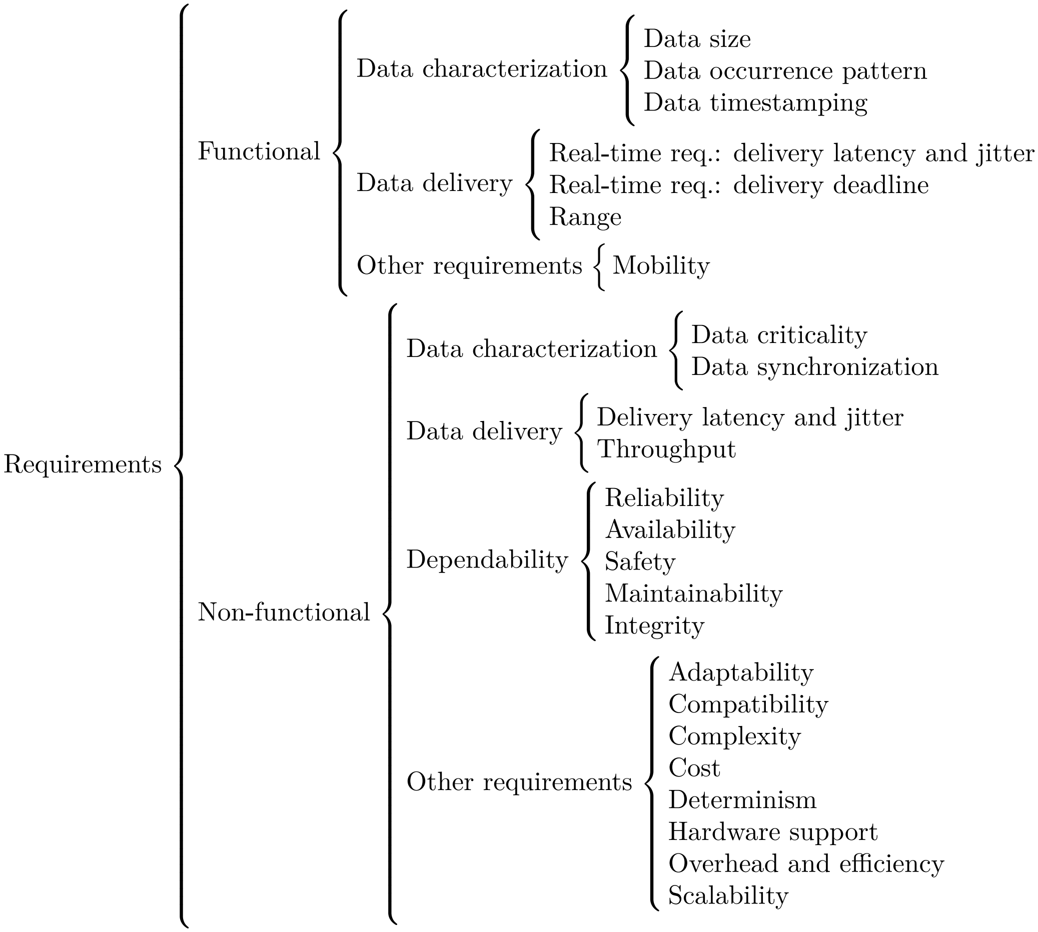


Figure 3.1: A taxonomy of communication requirements

Data size

Related terms	Payload
Functional/non-functional	Functional
Subject to optimization (Y/N)	No
Performance metric (Y/N)	No
Characterization	Data units

Computers are discrete systems in which events such as data transfers occur at a pace determined by the ruling clocks distributed across the communication system. Hence, data exchange is not expected to be continuous as in a broadcasting radio sender but in portions sent at discrete time intervals. In the communication stack, each layer has a limit to the amount of data, i.e., the payload, that the upper layers can process at once. Different applications may have different data size requirements, e.g., sensor data which is representing a physical magnitude can typically be encoded in a few bytes. In contrast, images captured by a camera are at least a couple of orders of magnitude larger. The data limit that each layer can handle is chosen as a compromise between the different impacted requirements. If an upper layer wants to send more data than the limit, the request can either be rejected or processed by splitting the data into several portions that do not exceed the given limit. The portions are then sent separately and reassembled at the destination. Depending on the conventions of the processing layer, these portions are given names such as packets, messages, frames or datagrams.

Data occurrence pattern

Related terms	Periodicity
Functional/non-functional	Functional
Subject to optimization (Y/N)	No
Performance metric (Y/N)	No
Characterization	• Periodic: period in time units • Aperiodic sporadic: Minimum inter-arrival time (MIT) in time units or rate expressed as data exchanges per time unit. • Aperiodic unbounded: not characterized

Modeling the pattern of how applications hand over data to the communication system is crucial for dimensioning the system's capabilities and allocating resources. A data exchange can be requested from the communication system periodically or aperiodically, hence the term periodicity is sometimes used. Aperiodic requests can in turn be classified as sporadic or unbounded. A brief explanation of these three categories is provided below, but a detailed description including their characterization is given later in the event handling characterization section (Section 4.1.1).

- Periodic. The data exchange is activated by a clock. Different clocks may be involved in the communication process, e.g., the sender's clock and the receiver's clock, and it may be important to determine whether these clocks are synchronized. If synchronization is not in place, a periodic event is perceived as drifting from the observer reference. Therefore, the following cases are distinguished:

  – Periodic not drifting. The data exchange is requested periodically and the clocks of the application that generates the data and the communication system are synchronized.

  – Periodic drifting. The data exchange is requested periodically and the clocks governing the application that generates the data and the communication system are not synchronized.

- Aperiodic sporadic or simply sporadic. The data exchange is requested at arbitrary times, but a maximum number of requests per time unit can be specified.

- Aperiodic unbounded or simply unbounded. The data exchange is requested at arbitrary times and it is not possible to specify an upper bound for the number of requests per time unit.

Data criticality

Functional/non-functional	Non-functional
Subject to optimization (Y/N)	No
Performance metric (Y/N)	No
Characterization	A quantifier that allows to distinguish between different criticality levels and their relative order.

The criticality of data refers to the impact of failure to meet communication requirements on the essential functions of the systems in which data is to be received and used. Assigning a criticality level to the data makes it possible to prioritize access to communication resources in situations where the system availability is not sufficient to meet all the requirements of the requested data exchanges. This means that the realization of requirements corresponding to a critical data exchange must be guaranteed over those with a lower criticality. Criticality levels must be provided taking into account the specifics of the respective use case. The criticality levels are usually linked to specific performance degradation levels in the target system. For example, from a fully functional system with all the criticality levels operating, to a system with a degraded performance in which only essential functions associated with highly critical functions are satisfied. A broadly applicable classification of the criticality of data is based on the following criteria:

- High criticality. Unfulfilled communication requirements can lead to severe system malfunction.

- Medium criticality. Unfulfilled communication requirements can lead to an impairment of operation, but not to a malfunction of the system.

- Low criticality. Not relevant for essential system operation.

A common confusion surrounding criticality concerns its relationship to the data occurrence pattern. It is often assumed that only periodic data exchanges are critical, while aperiodic and especially unbounded data exchanges are not critical. This assumption stems from the fact that periodicity makes the behavior predictable, making it more suitable for guaranteeing the requirements. At the same time, aperiodic events are not as predictable and their guarantees are less stringent. However, data exchanges with an unbounded activation pattern can also be critical. Therefore, the system may need to prioritize them over other, less critical periodic or sporadic data exchanges.

A review of the data communication technologies used in the 1990s and 2000s shows that criticality was a major reason for the division of communication technologies into different application areas, as in the case of IT and OT, which are discussed in detail in Chapter 5. The options were divided into those serving critical and non-critical use cases, with hardly any technology covering both. The gap arose in part from the initial needs of use cases that required one or the other, but did not prioritize covering both. The gap is also due to the challenges of having non-critical data exchanges in a system where criticality is compulsory. Such coexistence can raise the question of how to achieve isolation between the two. Systems that require both options, such as the control and entertainment system in an airplane, typically deploy two communication technologies. Alternatively, systems where two or more levels of criticality are handled are referred to as mixed-criticality systems [37].

Data timestamping and synchronization

Functional/non-functional	• Data timestamping: functional • Synchronization: non-functional
Subject to optimization (Y/N)	• Data timestamping: yes • Synchronization: yes
Performance metric (Y/N)	• Data timestamping: yes • Synchronization: yes
Characterization	• Data timestamping: Precision in time units. • Synchronization: Duration in time units to indicate the precision required by the synchronization.

The communication system may need to add a timestamp denoting the occurrence of an event related to the data, e.g., the time at which the data is generated or transmitted. The timestamp that is sent together with the data is interpreted by the receiver. Timestamps that refer to the time of data generation can be used, for example, to sort data by age or to discard data that exceeds a certain age limit. Timestamps are particularly interesting for applications that are installed in dynamic environments and whose sensors provide inputs that reflect the state of the environment. In such cases, old data refers to a former state and a reaction based on it is worthless. Data timestamping is usually more accurate when supported by hardware, but it is also possible to use only software-based solutions.

The use of timestamps brings another aspect into play: Timestamping events is not practical if the computer system that creates the timestamp and the computer system that interprets it are not synchronized, i.e., they do not have the same notion of time. Therefore, a synchronization protocol is required when calculations are performed with timestamps from systems that work with different clocks. The performance of a synchronization protocol is usually measured by its precision, i.e., the maximum deviation between two clocks at any point in time once the clock synchronization protocol is in operation and has stabilized.

The data timestamping precision and the synchronization precision are important performance metrics for systems that rely on a common time base and can be improved through optimization.

3.2 Data delivery requirements

The requirements grouped under data delivery refer to how the data is to be handed over to the receiver by the communication system.

Real-time requirement: delivery latency and jitter

Related terms	Response time, channel access delay, delivery delay.
Functional/non-functional	• Real-time applications: functional • Non-real-time applications: non-functional
Subject to optimization (Y/N)	Yes
Performance metric (Y/N)	Yes
Characterization	Duration in time units

The delivery latency measures the time that elapses between the moment when data is handed over to be sent by the communication system until it is available at the receiver. Other related terms are response time, which is the time it takes to handle a data transfer, and channel access delay, as the time it takes to effectively access the channel is often an important factor in delivery latency. Delivery latency is a crucial requirement and performance parameter for real-time applications. It can be set as a value to optimize since its reduction is normally beneficial, e.g. to decrease the probability of missing a delivery deadline.

This requirement can be further refined and expressed in terms of bounded or guaranteed latency (Figure 3.2). A bounded latency refers to a request where, in the worst case, the data must be delivered with the specified latency value, but can also be delivered earlier. This is usually linked to a delivery deadline that cannot be exceeded. The utility of data is therefore very often tied to the time at which it is received, with earlier delivery being desirable. The case of guaranteed latency, on the other hand, means that the data is delivered consistently with a constant latency value that only allows very small fluctuations, i.e., has a very low delivery jitter. An example of bounded latency is audio and video streaming traffic, where data may arrive earlier as long as it is available before being reproduced. In contrast, control applications may require latency to be guaranteed as control loops must respond to data inputs in a tight time frame.

The delivery latency can be influenced by many factors that play a role in the transmission of data. It is therefore often defined as a value for optimization and as a performance metric.

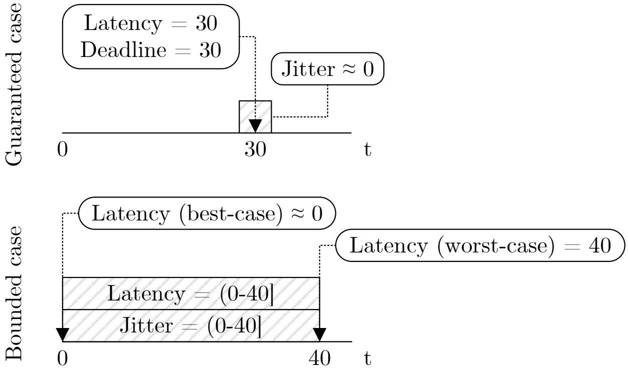


Figure 3.2: Data delivery with guaranteed or bounded latency

Real-time requirement: delivery deadline

Functional/non-functional	Functional
Subject to optimization (Y/N)	• Hard deadlines: no • Soft deadlines: yes
Performance metric (Y/N)	• Hard deadlines: no • Soft deadlines: yes
Characterization	Duration in time units (relative value) or point in time in time units (absolute value).

The delivery deadline is another crucial requirement for communications serving real-time applications. The delivery deadline can be expressed as an absolute value, i.e., as a duration, or as a relative value, i.e., as a point in time. If the absolute value is used, the deadline is likely to start at the moment the data is made available to the communication system. If the data occurrence pattern is periodic, the deadline is often set to match the time period so that the data exchange is completed before a new instance of the data is available. Essentially, the deadline sets a limit for the delivery latency, beyond which the usefulness of the data decreases or it even becomes unusable.

On the receiving end of a communication system, deadlines are set so that a chain of computations and communication steps concerning the data can be met without surpassing a defined delay. In fact, sometimes the deadline for an entire chain of events is more important than the deadline for the individual steps. For example, a system performs some computation before transmitting the results to a receiving system that processes the data. In such cases, the deadline of some individual steps (in the given example, the sequence is computation, communication and computation) may be exceeded as long as the deadline of the entire chain is met (in the given example, the result of the last computation is obtained).

Deadlines can also be characterized as hard or soft, depending on the consequences of exceeding them. A deadline is hard if the data transfer becomes unusable if it is exceeded. A deadline is soft if the data is still valid after the deadline has been exceeded, even if this validity is expected to decrease over time.

If the applicable deadlines are hard, they cannot be selected as optimization or performance metrics, as they are mandatory. On the other hand, if they are soft deadlines, the reduction in the number of missed deadlines could be set as an optimization or performance measure.

Throughput

Related terms	Bandwidth, rate
Functional/non-functional	Non-functional
Subject to optimization (Y/N)	Yes
Performance metric (Y/N)	Yes
Characterization	Data units per time units

Throughput, often referred to as bandwidth or rate, relates to the amount of data transferred per unit of time. Such a notion can also be derived from the combination of the data size and occurrence pattern requirements mentioned above. Nevertheless, throughput often fails to express details about how frequent and how much data should be exchanged. Its definition is tied to a time window and does not clarify whether the data comes in bursts, is evenly distributed or follows any other pattern. For example, the available throughput may meet the average needs of applications exchanging data over time, but the applications may request it according to a pattern that cannot be met by the supplied throughput during certain time intervals. Figure 3.3 shows an example of such a situation.

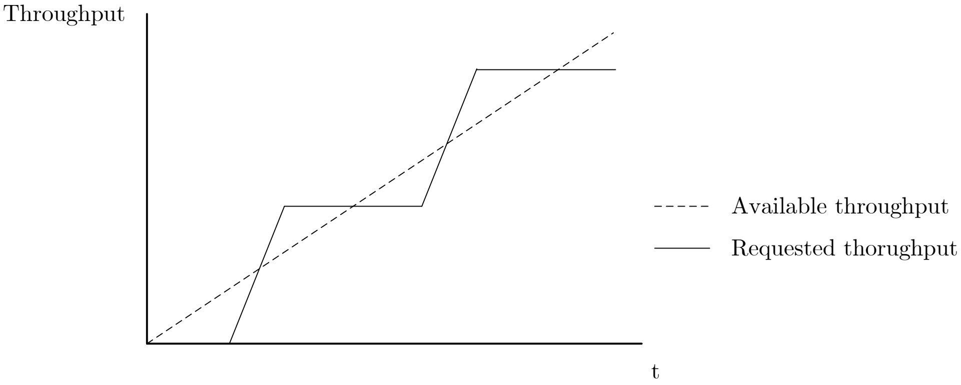


Figure 3.3: Requested throughput vs. available throughput. The average throughput for the entire period is the same in both cases, but the requested throughput is not reached in some time intervals.

Specifying the throughput that the communication technology should fulfill is often made to provide an upper value which the actual data transmissions will not reach or will reach only in the worst-case scenarios. Nevertheless, the throughput can be quite descriptive in systems where there are no fluctuations in the amount and pace of data exchanged.

Throughput can be used as a performance metric and optimization target, but is often considered less relevant compared to the fulfillment of dependability and real-time requirements.

Range

Functional/non-functional	Functional
Subject to optimization (Y/N)	No
Performance metric (Y/N)	No
Characterization	Distance units

Range refers to the distance over which the communication technology can transfer data between a pair of sender and receiver. It is largely determined by the transmission media and factors such as the frequency band and signal power. In most cases, these factors can be overcome by the use of intermediate devices that relay the data and extend the coverage. Therefore, range is a decisive criterion for communication solutions, which are often classified according to it, e.g., LAN or PAN. In some cases, the range of wireless deployments is extended by using hybrid wired and wireless solutions. For example, several wireless nodes could be used as intermediate devices in different areas. These nodes also have a wired interface and are all connected via a wired network, allowing data exchange between the different areas. Other wireless nodes could move between the areas covered by these intermediate nodes, with the transfer between areas commonly referred to as a handover.

The range is usually not selected as a performance metric or as a target for optimization, but as a compulsory requirement to be met by the network deployment.

3.3 Dependability requirements

Dependability offers a framework that encompasses various attributes that vary slightly depending on the author's point of view, each offering a different perspective that better suits one scenario or another. The definition from [9] is chosen as it summarizes valuable attributes for this thesis: reliability, availability, safety, maintainability and integrity. Dependability is discussed further in the dependable systems section (Section 4.2). Just as dependability is defined as a composite of several attributes, [9] defines security as a composite of confidentiality, availability and integrity, where confidentiality means that no user accesses data that is not authorized to have access to. Although security is a key aspect in many deployments, it is outside the scope of this thesis and is therefore not discussed in more detail apart from the descriptions of the attributes of availability and integrity due to their impact on dependability.

Reliability

Related terms	Tolerance to loss, tolerance to interference, packet error rate.
Functional/non-functional	Non-functional
Subject to optimization (Y/N)	Yes
Performance metric (Y/N)	Yes
Characterization	Proportion of data exchanges that fulfill the functional requirements out of the total number of requested data exchanges during a period of time, expressed in units of time.

The reliability of a communication system indicates the extent to which data exchange is carried out in accordance with the functional requirements, i.e., communication is performed the way it is expected, over a time period of time. It relates the number of times the data exchange is accomplished with the number of times the data exchange is requested within a time interval. If one or more of the functional requirements are not met, reliability gets compromised. Other terms related to reliability are packet error rate, which is inversely proportional, and tolerance to loss or interference, since a system that is able to handle these impediments is more reliable.

Since a time duration is used in the formulation of reliability, its value is subject to the selected time window (Figure 3.4). For example, a communication system might suffer from bursty interference and the reliability value is zero if it only refers to a time period in which the interference burst occurs. On the other hand, if a larger time window is considered, the bursts may not be as relevant. Therefore, it is important to properly evaluate the scenarios in which the communication system is deployed and their outages to determine whether data exchanges will be compromised.

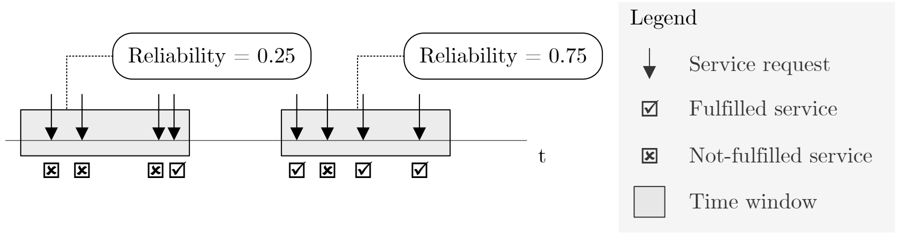


Figure 3.4: Fluctuations in reliability depending on the selected time window.

Furthermore, it is not realistic to set an actual value of 100% for reliability, as there is always a chance, albeit small, that the system will fail. Nevertheless, the communication system could still function properly even if the functional requirements are not met 100% of the time. For example, if the communication system only fulfills the functional requirements 80 to 95% of the time, the entire system can be designed to work as long as communication is successful at least 80% of the time. Nevertheless, the temporal distribution of the 20% of unsuccessful data exchanges is also crucial, as already mentioned in the reliability windows explanation. For example, clock synchronization algorithms often work under this assumption. If some of the data exchanges they perform to synchronize the systems are lost, the quality of synchronization deteriorates, but may still be within an acceptable threshold. Also, a reliable system is likely to fail after some time due to overheating, electrical damage or physical wear.

Reliability can be selected as a target for optimization and is often an important performance parameter, especially for systems hosting critical applications.

Availability

Related terms	Responsiveness
Functional/non-functional	Non-functional
Subject to optimization (Y/N)	Yes
Performance metric (Y/N)	Yes
Characterization	Proportion of time in which the system is ready to fulfill the functional requirements of data exchanges over a selected period of time, expressed in time units.

The availability of a communication system indicates the extent to which data exchange can be offered in accordance with the functional requirements over the requested period of time. A related term that conveys the same idea is responsiveness. The reasons why a system is unavailable are due to failure to meet requirements during operation or maintenance of the system. As with reliability, availability is defined over a period of time and is therefore subject to fluctuations depending on the temporal location and size of the time window under consideration.

Availability is usually considered together with reliability as an optimization target and is often a relevant performance metric.

Safety

Functional/non-functional	Non-functional
Subject to optimization (Y/N)	Yes
Performance metric (Y/N)	Yes
Characterization	No concrete way to specify safety. Some metric could be used to represent safety, providing a number or set of numbers that serve as a relative value to compare the performance of different implementations and in different scenarios.

Safety is about the absence of faults that could cause harm to users or the environment. The extent of the consequences of communication not meeting functional requirements depends on the applications and also on how much they rely on the communications to perform critical tasks that could lead to harm. Safety measures aim to reduce the risks of situations in which communication is faulty or to minimize their consequences.

Safety measures can be part of the optimizations and safety could be one of the metrics used to evaluate performance.

Maintainability

Functional/non-functional	Non-functional
Subject to optimization (Y/N)	Yes
Performance metric (Y/N)	Yes
Characterization	No concrete way to specify maintainability. Some metric could be used to represent maintainability, providing a number or set of numbers that serve as a relative value to compare the performance of different implementations and in different scenarios.

Maintainability describes the extent to which a system can be kept functional while being subject to repairs and modifications.

Integrity

Functional/non-functional	Non-functional
Subject to optimization (Y/N)	Yes
Performance metric (Y/N)	Yes
Characterization	No concrete way to specify integrity. Some metric could be used to represent integrity, providing a number or set of numbers that serve as a relative value to compare the performance of different implementations and in different scenarios.

The attribute integrity refers to the absence of system alterations, which in a communication system primarily refers to the data being exchanged with completeness, accuracy and absence of unauthorized modifications. A distinction can be made between the reasons why data is being altered. On the one hand, issues related to the communication channel, such as interference or fading, for which the receiving system may attempt to recover or discard the data. On the other hand, when data appears to have a legitimate format and content, but the information has been maliciously altered prior to receipt.

3.4 Other requirements

Other requirements that do not belong to the previously introduced categories are presented below.

Adaptability

Related terms	Flexibility
Functional/non-functional	Non-functional
Subject to optimization (Y/N)	Yes
Performance metric (Y/N)	Yes
Characterization	No concrete way to specify adaptability. Some metric could be used to represent adaptability, providing a number or set of numbers that serve as a relative value to compare the performance of different implementations and in different scenarios.

Adaptability or flexibility refers to the ability to cope with changes in communication requirements that occur dynamically while the communication system is in operation. For example, it may be necessary to adapt to changing channel conditions. Another example might require the communication system to cope with the addition or removal of applications that exchange data while the system is in operation. In this situation, the potential adaptations need to be evaluated to ensure that they do not affect the existing system, e.g., via schedulability tests to figure out if the real-time requirements can be met. In addition, the changes might be difficult to manage at runtime, e.g., if the communication system relies on a specific configuration that cannot be easily derived in a short time to respond to the changes at runtime.

Compatibility

Functional/non-functional	Non-functional
Subject to optimization (Y/N)	Yes
Performance metric (Y/N)	Yes
Characterization	No concrete way to specify compatibility. Some metric could be used to represent compatibility, providing a number or set of numbers that serve as a relative value to compare the performance of different implementations and in different scenarios.

Communication deployments are often a collection of multiple communication technologies, often designed independently of each other, but with an expected level of compatibility. In some cases, the technologies are responsible for different tasks in the communication architecture and are combined to achieve the goal of the communication system. For example, Ethernet and IP are often used together even though they are independent technologies. In other cases, the technologies cover similar tasks in the communication architecture, but are used in different application fields. For example, Ethernet and IEEE 802.11 perform the tasks of the physical and data link layer of the communication architecture, but are used in wired and wireless deployments, respectively. Sometimes deployments such as those based on Ethernet and IEEE 802.11 are arranged in a way they complement each other and can exchange data. Such scenarios pose a particular challenge as the solution offered by each of these technologies may be designed to tackle distinct requirements. This might force dealing with aspects such as different data sizes, throughput or reliability levels, to name but a few. In other cases, applications are competing for the same resources, e.g., use the same transmission frequencies, and special care must be taken to minimize the disturbance caused to each other and allow coexistence. Very often the set of communication technologies is predetermined, there may also be legacy systems, and the solutions must take such constraints into account and adapt to them.

The level of compatibility can be set as a performance metric and serve as a target for optimization.

Complexity

Functional/non-functional	Non-functional
Subject to optimization (Y/N)	Yes
Performance metric (Y/N)	Yes
Characterization	No concrete way to specify complexity. Some metric could be used to represent complexity, providing a number or set of numbers that serve as a relative value to compare the performance of different implementations and in different scenarios.

The complexity of the communication solution refers to the difficulty of fulfilling the requirements. There may be several design options that can provide good coverage for the given set of requirements. When selecting an option, other factors besides complexity must be considered, such as cost, or overhead and efficiency. Complexity is behind the adoption of layered architectures for communication systems. In this way, each layer can take responsibility for specific functionality and hide its complexity from other layers.

Performance metrics can be set to measure complexity and its fulfillment can be selected as an optimization goal.

Cost

Functional/non-functional	Non-functional
Subject to optimization (Y/N)	Yes
Performance metric (Y/N)	Yes
Characterization	Time units or money units.

Communication solutions incur costs in the form of time and money invested in development, deployment and maintenance. Each of these phases contributes to the overall cost to varying degrees, with investments in one phase potentially having a positive impact on another. For example, the use of already available solutions based on COTS communication devices can be expected to reduce development and deployment costs, as these are based on widely used and readily available technologies, unlike ad-hoc solutions. However, COTS are not always applicable as their design may not take into account some of the demanded requirements.

Determinism

Related terms	Predictability
Functional/non-functional	Non-functional
Subject to optimization (Y/N)	No
Performance metric (Y/N)	No
Characterization	Boolean indicating if determinism is required or not.

Determinism indicates whether the state of the system is unambiguously given on the basis of the previous state. This means that the future state is determined by the past state. It is often mentioned together with predictability, which refers to the ability to figure out what will happen in the system based on previous events. Although absolute certainty is unrealistic, a system designed to reduce uncertainty can reach the expected state with a high degree of confidence based on the previous state and available transitions. Therefore, limiting the number of possible states or defining clear transitions between them can improve the determinism of the system. For example, the different layers involved in the communication stack may be subject to random processes which result in different timing. In such a scenario, a time interval could be defined for each of these processes so that the probability of them completing within that interval is high enough to execute the entire sequence of events with high confidence. Similarly, transmission failures can occur in the communication system due to poor channel quality. However, if these problems are anticipated, the communication system may be able to cope with them, making the state of the system known. For example, the so-called schedulability tests are helpful analysis tools that help answer the question of whether the available processing and communication resources can fulfill the required timing communication requirements.

The determinism of a system can be defined in different domains or different levels of abstraction. In the case of a communication system, for example, determinism could be defined by the fact that it is possible to know at any time whether the system is in one of the following three states: transmitting, not transmitting or failing to transmit. However, from a safety perspective, determinism could only refer to whether the system is in a safe state or not, leaving out the details about the state of transmissions.

The determinism of the system is a boolean attribute: The system is either deterministic or not. It is not intended as a target for optimization or as a performance metric. However, the predictability of the system can be improved and set as a goal for optimization.

Hardware support

Functional/non-functional	Non-functional
Subject to optimization (Y/N)	No
Performance metric (Y/N)	No
Characterization	Boolean indicating if a specific hardware implementation is required or not.

The implementation of a communication solution relies to varying degrees on the hardware to fulfill its task. In the physical layer, for example, the transmission of signals over the medium requires hardware support. In recent years, there has been a trend to shift former hardware-based responsibilities to software, especially in general-purpose systems, where the flexibility and power of the platform allow for complex solutions to be carried on. This is the case with software-defined radios (SDRs), where the physical layer modules that were traditionally implemented in hardware are now handled in software. Some chipsets also allow part of the MAC to be controlled in software, originating a distinction between the MAC protocols that are handled entirely in hardware (full or hard MAC) and those that are at least partially handled in software (soft MAC). Other layers may require a mixture of hardware and software, but the upper the layer, the more likely it is that its implementation will be done in software. Overall, the decision to choose specific hardware or software tasks depends on many factors, and a case-by-case analysis is required.

Mobility

Functional/non-functional	Functional
Subject to optimization (Y/N)	No
Performance metric (Y/N)	No
Characterization	Boolean indicating if mobility is required or not.

The systems requiring connectivity might be subject to different degrees of mobility. When nodes are static, a wired deployment is often the most sensible option. In other cases, e.g., in the presence of robotic arms, connectivity may be required between moving nodes, which could require the removal of cables depending on the characteristics of the movement. When full mobility is required, e.g., for a transportation robot in a factory, wireless deployments are the only option. As previously explained for the range requirement, some wireless deployments offer an extended range supported by multiple intermediate nodes. A wireless node that requires a high degree of mobility may move from the range of one intermediate node to another, a process that may require a handover procedure to keep the communication. Mobility can also cause wireless channel conditions to change over time as the location of nodes and other influencing factors such as obstacles change.

Mobility is not expected to be measured as a performance metric or set as a target for optimization, but rather as a feature that is either required or not. If required, it can be expressed, for example, by the distance that nodes must be able to travel.

Overhead and efficiency

Related terms	Performability
Functional/non-functional	Non-functional
Subject to optimization (Y/N)	Yes
Performance metric (Y/N)	Yes
Characterization	No concrete way to specify overhead and efficiency. Some metric could be used to represent overhead and efficiency, providing a number or set of numbers that serve as a relative value to compare the performance of different implementations and in different scenarios.

The communication system suffers from inefficiencies in the data exchange that do not allow the theoretical throughput limit to be reached and constrain the performance of the system. The reasons for such inefficiencies are manifold. For example, the use of relatively large headers in data exchange to support the functioning of communication protocols becomes a problem when small amounts of data are exchanged. Supporting protocols such as clock synchronization also consume some of the available bandwidth, but may be necessary to transfer data according to requirements. Another example is related to how the protocols sort out the access to the medium. For example, designing a protocol with pre-assigned schedules avoids to send a polling message from a coordinator before each data transfer, saving some bandwidth. A key source of inefficiency is the communication layers and the split of responsibilities between them, which often results in layers that are more manageable but not fully coordinated. Such a lack of coordination can lead to longer processing times, for example.

In addition, other resources such as energy play a crucial role to the extent that some communication technologies are fully designed to achieve energy consumption targets. For example, some technologies are deployed in scenarios with energy constraints and are powered by batteries that cannot be easily replaced.

Scalability

Functional/non-functional	Non-functional
Subject to optimization (Y/N)	Yes
Performance metric (Y/N)	Yes
Characterization	No concrete way to specify scalability. Some metric could be used to represent scalability, providing a number or set of numbers that serve as a relative value to compare the performance of different implementations and in different scenarios.

The scalability of the communication system refers to how well it reacts to the addition of new applications or the extension of its capabilities. That is, whether the new applications and extensions can be successfully added without compromising the existing system. For systems with timing requirements, the set of available communication resources may need to be allocated in favor of determinism, i.e., to reduce uncertainty. In this context, schedulability tests are often used to evaluate how well the system scales in terms of the assignment of computing and communication resources. Common limiting factors for scalability are the throughput or the topologies used to connect the communication peers. In addition, latency requirements may be more difficult to meet in a more extensive and congested communication system. Another limitation is the characteristics of the medium, e.g., the presence of overlapping channels that cannot be used simultaneously.

3.5 Fulfillment of communication requirements: the use of traffic classes

Some communication systems fulfill the requirements of applications with dissimilar needs by offering different service tiers instead of providing the same service to all applications. Hence, when deploying communication systems, it is often the case that different requirements are grouped together and served by different so-called traffic classes. These classes are often realized by different software and hardware mechanisms.

Some of the requirements commonly used in defining traffic classes are the occurrence pattern they serve, the timing requirements they can fulfill or the criticality level for the data they carry. Traffic classes often define service tiers with varying degrees of fulfillment of requirements, from more relaxed to more demanding. For example, one traffic class could be defined for data transmissions with no delivery latency requirements, another for transmissions with a bounded delivery latency and a third for transmissions that require a guaranteed latency.

In certain systems, the requirements of the applications cannot be fully mapped to the available traffic classes. In such cases, it is necessary to assign the application to a traffic class that might be meant to fulfill more demanding requirements, with the disadvantage that the capacity of such a traffic class is not used efficiently. In the example previously mentioned, this would be the case if a data transmission requiring bounded latency is ultimately handled by a traffic class supporting guaranteed latency.

Examples of different traffic classes can be found later in the review of communication technologies in Chapter 5.

Chapter 4

Specialized systems: Addressing real-time and dependability requirements

The review of communication requirements in Chapter 3 has shown the importance of real-time and dependability requirements, especially in scenarios related to the OT domain. The demanding nature of these requirements requires systems specifically designed to handle them. This chapter provides a detailed description of systems supporting real-time and dependability requirements, covering topics such as scheduling, fault taxonomy or threat handling, just to name a few, which will be used extensively in the contributions part (Part II).

4.1 Real-time systems

The data processing in computer systems and the data exchange in the communication networks often require the consideration of timing requirements. The real-time system continuously monitors the environment to retrieve data, process it and generate a response that often interacts with the environment (Figure 4.1). Specifically, the environment stimulates the system via sensor inputs and the system generates a set of outputs to control actuators. The actuators then have an effect on the environment. These closed-loop systems perform some computations where not only a result must be delivered, but also the time at which this happens.

Real-time systems usually rely on embedded computer systems to fulfill their job. Thus, this thesis is interested in the distinction between general-purpose systems and embedded systems among other taxonomies that can be used to classify computer systems. The capacity of general-purpose systems in terms of computing power, memory, storage or interconnectivity is such that they can fulfill a wide range of tasks pursuing a variety of goals. Examples of general-purpose systems are an Internet server or a PC. In contrast, embedded systems are computers that are characterized by a specific function that they are intended to perform. The reader has probably heard countless times how ubiquitous computers are and that everyday devices easily contain several of these embedded computer systems. Due to their strong interaction with the physical environment and real-time requirements, the term cyber-physical systems has been coined to describe this type of embedded systems. An example of a cyber-physical system is the anti-lock braking system (ABS), which prevents vehicles from blocking their wheels completely, a situation that could lead to a loss of control of the vehicle. An ABS is expected to detect the blocked wheels and partially release the brakes while attempting to reduce the speed of the vehicle, which is the ultimate reason for a driver to apply the brake pedal. Although the distinction between general purpose systems and embedded systems seems to be dictated by hardware, software plays a crucial role in accomplishing the goals. Often a mixture of hardware and software is used to fulfill the goals of the different scenarios.

The formulation of real-time requirements serves to determine whether the data processing and communication steps that may be triggered in the system in response to a stimulus should be completed within certain latency and jitter values or before a deadline (as described in the data delivery requirements in Section 3.2). In real-time systems, the validity of the response is therefore maximized in the time windows specified by the timing requirements. If the response is completed earlier or later than specified in the required time window, the utility of the response ranges between limited to useless.

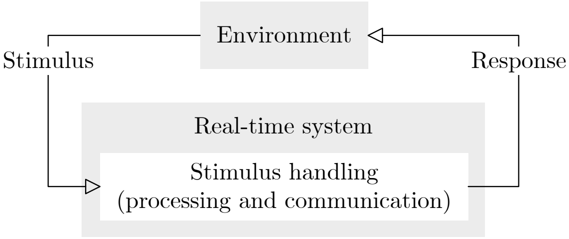


Figure 4.1: Monitoring, handling and response generation to a stimulus in a real-time system

Real-time systems do not necessarily need to be fast, but they do need to be predictable. A fast system may have more margin to meet timing requirements such as deadlines, but does not serve the purpose if these deadlines are met arbitrarily across multiple instances of stimulus handling.

4.1.1 Event handling characterization

Data subject to be processed or exchanged stimulates the processing and communication systems either periodically, when activated by a clock, or aperiodically, like many of the interrupts issued by hardware devices (see data occurrence pattern requirements in Section 3.1). The arrival of a stimulus in the system is commonly referred to as an event. The processing of events is done by tasks, a term commonly used in the real-time and operating systems domain that refers to the method required to handle the stimuli. Other terms such as threads or processes are also used to refer to the hardware or software entities that perform the work, with subtle variations in the meaning.

Periodic events can be guaranteed to be handled by the processing and communication systems since all the information required to manage them is known at the time of designing the system. Consequently, resources like computing time, memory and communication bandwidth can be allocated to handle the events so that the timing requirements for their response are met. However, a problem may arise if the clock that generates the stimuli and the clocks in the processing and communication systems are not synchronized. Although the stimuli may periodically provide data to the handling system, the handling system could perceive that the pace of the stimuli is drifting. For example, processing time could be assigned to handle a periodic sensor input in a computer system. If the clocks of the sensor and the computer system are synchronized, the processing time could be scheduled with the same period that the sensor generates data, immediately after the sensor data is generated, so that the data is processed in the shortest possible time. However, a lack of synchronization between the sensor and the system makes it impossible to attain the shortest processing time. In the case of aperiodic events, the time of their occurrence is not known, thus they are characterized by their variance. The presence of aperiodic events is expected, e.g., in systems with human interaction or in systems that process interruptions from hardware devices. Depending on the amount of information available from aperiodic events, it is possible to guarantee their handling if they are sporadic. This means that the maximum rate at which the event occurs is bounded. For example, the number of times a key on a keyboard is pressed is constrained by physical limits. If there is no information about possible boundaries for the occurrence of an event and events occur in the system, resources can only be allocated for limited predictability.

Baseline temporal characterization

A task τ_i or data transmission m_i is first characterized by the execution or transmission time C_i, respectively (Figure 4.2). The allocation of computing and communication resources must be done in such a way that the presence of other tasks and data transfers does not prevent the fulfillment of the requirements, a job that is the responsibility of the scheduling algorithm. The guarantees given by the scheduling algorithm are based on the consideration of C_i for the execution or data delivery time. The execution time C_i is usually set to as worst-case execution time (WCET). The WCET or data delivery time estimation can be complicated, especially for systems with a high level of complexity where plenty of factors can influence the worst-case scenario. In general-purpose computer systems, for example, many sources of non-determinism complicate the estimation of WCET, including delays introduced by memory cache misses or variability in the execution time of operating system calls. While using an operating system provides access to a wide range of services such as communication, concurrency handling or execution scheduling, it introduces uncertainties. In a communication system, transmission times vary due to aspects such as rate control algorithms adapting to varying levels of channel noise. In such cases, the WCET or data delivery time estimates can be so pessimistic that the system is no longer feasible. Very often, the WCET or data delivery time is determined after running the tasks and performing the transmissions multiple times until having enough confidence in the results.

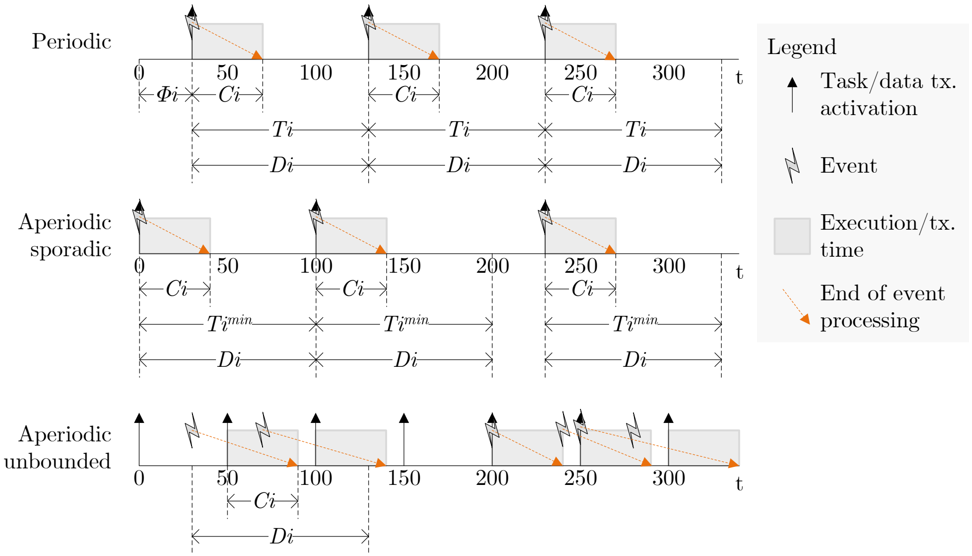


Figure 4.2: Baseline characterization of a task τ_i or a data transmission m_i

The characterization of τ_i or m_i in real-time systems requires the specification of a deadline D_i, which denotes the moment when the handling of the stimulus shall be completed. The deadline is set relative to the time at which the stimulus appears in the system. Real-time systems are classified into hard and soft systems, depending on the consequences of overrunning the deadline, as explained in the definition of the delivery deadline requirement from Section 3.2. A system is hard real-time when the failure to handle an event on time makes the response useless and soft real-time when the validity of the response decreases but is still useful once the deadline is reached.

A periodic τ_i or m_i is additionally defined by the specification of its period T_i. Furthermore, the offset Φ_i is used to delay the occurrence of the triggering event with respect to the starting time 0 at which it is otherwise assumed that all tasks or data transmissions are activated. Periodic events are often set to be handled before a deadline equal to the period, ensuring that the new instance of the periodic event occurs when the previous event processing has already been completed. All in all, the modeling of a periodic τ_i or m_i is defined by the tuple (C_i, D_i, T_i, Φ_i).

The occurrence pattern of an aperiodic sporadic or simply sporadic τ_i or m_i is defined either by the frequency f_i or the MIT T_i^{min}. The frequency or rate indicates the maximum number of events to be handled per time unit. However, it says nothing about the distribution of events during the given time window. Such a definition could lead to situations that lie between two extremes. On the one hand, the stimuli are concentrated in one part of the time window. On the other hand, the stimuli are evenly distributed over the given time. The processing or communication resources handling such stimuli often have to be prepared to deal with both cases. With T_i^{min}, the shortest time that must elapse between each pair of consecutive stimuli is specified. This specification guarantees that the handling system does not have to manage stimuli that appear more frequently than defined by T_i^{min}. In the example from Figure 4.3, the timeline above depicts an event that occurs four times every 200 time units. However, during the first three task activations no events occur, which results in missing three chances to process the events. The events come later in a burst and do not have enough processing resources left. In the case of a definition using MIT, the example in the bottom part of Figure 4.3 shows how a guarantee is given that the events have a minimum time interval between them, which could help to better dimension and allocate the processing and communication resources. Thus, the modeling of a sporadic τ_i or m_i can be expressed by the tuples (C_i, D_i, f_i) or (C_i, D_i, T_i^{min}).

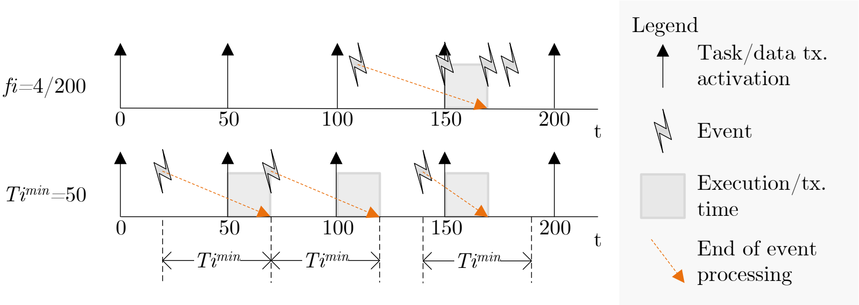


Figure 4.3: Specification of the sporadic occurrence pattern of a task τ_i or a data transmission m_i using the frequency f_i vs. the MIT T_i^{min}

In the case of an aperiodic unbounded or simply unbounded τ_i or m_i, their modeling does not include additional parameters since there is no information about their occurrence pattern. Therefore, an unbounded τ_i or m_i is fully modeled with the tuple (C_i, D_i).

Table 4.1 summarizes the baseline characterization of τ_i and m_i according to their occurrence pattern.

It should be noted that the pattern of generation of events does not always match the pattern of their handling. Table 4.2 gives an overview of the consequences of each of the possible combinations between the generation and handling stages.

Extended characterization

In addition to the temporal aspects that characterize the handling of τ_i and m_i events, the criticality X_i associated with the event may also have to be taken into account. If critical events are not handled, be they periodic, sporadic or unbounded, this can lead to malfunctions in essential systems. In turn, non-critical events could be neglected while critical events are handled, as the consequences of not fulfilling the requirements for non-critical events do not affect essential functions. The criticality of events relates to the notion of hard and soft real-time systems. A critical system is unlikely to be able to tolerate a missed deadline and is therefore classified as hard real-time system. In contrast, a system with low criticality is expected to fall into the category of soft real-time systems. The definition of criticality levels (see data criticality requirements in Section 3.1) depends on the system in question. The purpose of such a definition is to characterize tasks and data transmissions in order to provide different service levels.

Table 4.1: Baseline characterization of a task τ_i or a data transmission m_i

Task or data transmission occurrence pattern	**Characterization**	
Periodic τ_i or m_i	Execution/transmission time	C_i
	Deadline	D_i
	Period	T_i
	Offset	Φ_i
Sporadic τ_i or m_i	Execution/transmission time	C_i
	Deadline	D_i
	Option A:	
	Frequency	f_i
	Option B:	
	Minimum inter-arrival time	T_i^{min}
Unbounded τ_i or m_i	Execution/transmission time	C_i
	Deadline	D_i

Table 4.2: Consequences of the different combinations of event generation and event handling

Generation	**Handling**	**Comment on consequences**
Periodic (not drifting)	Periodic (not drifting)	Handling period equal to generation period. Minimized handling jitter.
	Sporadic	Sporadic handling with MIT equal to generation period incurs in handling jitter.
	Unbounded	No guarantees
Periodic (drifting)	Periodic (drifting)	Unsynchronized. Might cause unbounded handling jitter.
	Sporadic	Unsynchronized. Might cause unbounded handling jitter.
	Unbounded	No guarantees
Sporadic	Periodic (not drifting)	Periodic handling with period equal to generation MIT incurs in handling jitter.
	Periodic (drifting)	Unsynchronized. Might cause unbounded handling jitter.
	Sporadic	Handling MIT equal to generation MIT incurs in handling jitter.
	Unbounded	No guarantees
Unbounded	Any	No guarantees

Some of these classifications are meant to apply to different application fields, such as the Automotive Safety and Integrity Levels (ASIL) in the automotive industry or Design Assurance Level (DAL) in the aerospace industry. Scheduling algorithms often take the criticality into account in order to prioritize the processing of certain tasks and data transmissions over others [37]. Prioritization is particularly important in mixed-criticality systems, in which tasks and data transmissions with different criticality share limited computing and communication resources.

The response time R_i or delivery latency for τ_i or m_i (Figure 4.4) indicates the time that elapses between the occurrence of a task or a data transmission event and the completion of the handling of this event. It is an important performance parameter in real-time systems that indicates whether a task or data transmission can fulfill its timing requirements, in particular its deadline. The response time includes the execution time of the task or the data transmission time C_i and the effects caused by other tasks or data transmissions in the form of preemption and blocking. In systems with preemption, the handling a task or data transmission can be interrupted if the system has to serve another task or data transmission at that time. The preempted work is resumed at a later time, but can often go through several pause and resume cycles before it is completed, resulting in an accumulated delay. Preemption is a common feature in most modern general-purpose operating systems. However, it is often missing in communication networks, so that a transmission that has been started is usually completed before a new one is started. Suppose a task or message transmission is to be handled, but another task or message transmission is using the processing or communication resources. In this case, this is considered a blocking time and must also be taken into account when calculating the response time. The upcoming section deals with scheduling and provides a deeper insight into the analysis techniques used to calculate the response time.

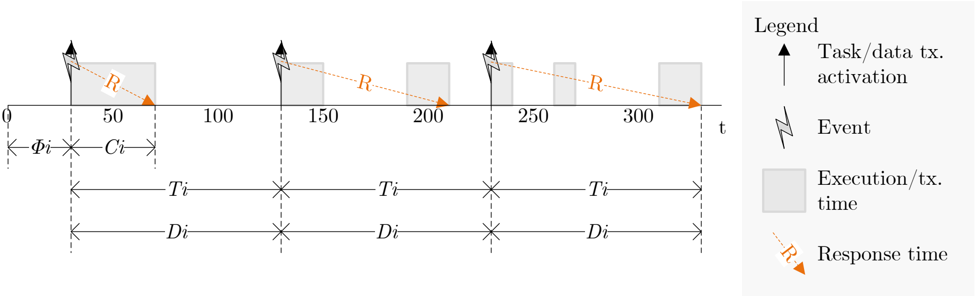


Figure 4.4: Response time of a task τ_i or a data transmission m_i

The different τ_i or m_i instances are likely to experience variations in their timing behavior expressed by the jitter term (Figure 4.5). Jitter defines the range of values that a time magnitude could take. Three types of jitter are relevant when dealing with τ_i or m_i. Firstly, the jitter in the release of the event to handle $J_i^{release}$. Although often overlooked, periodic events have unbounded jitter on release if the clock ruling the generation of the event and the clock in the handling system are not synchronized. Second, jitter manifested in the execution or transmission time $J_i^{handling}$. The variations in the duration of a processing task can be remarkable because a task executes a program that can easily execute conditional statements that direct the program flow along different paths that have different durations. Data transmissions might as well experience different transmission times among several instances. For example, the rate control algorithms regulate the throughput of a transmission to adapt to the noise values of the channel, so that transmissions of the same size could result in different transmission times. It is also a possibility that the amount of data that needs to be transmitted on each transmission instances varies, affecting the handling jitter. Another source of transmission time variability is time-diversity mechanisms to improve reliability and availability, as transmissions may have multiple transmission attempts before they are successful. The third type of jitter, the jitter in the response time $J_i^{response}$, is affected by the execution, blocking and preemption times.

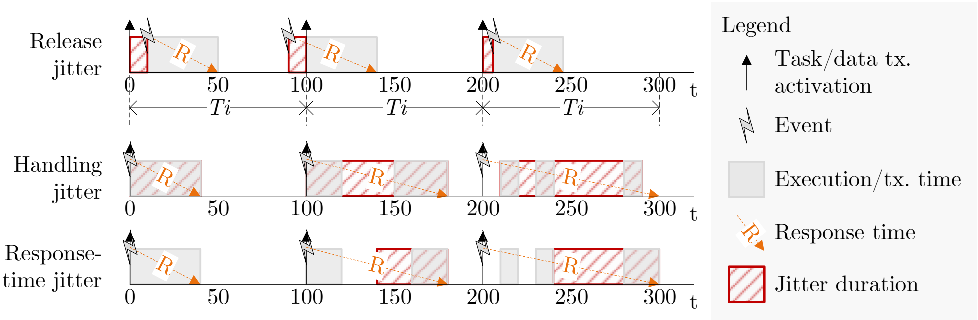


Figure 4.5: The different types of jitter that affect a task τ_i or a data transmission m_i.

The release jitter is often considered in the configuration of the system handling the event in order to optimize the response. Handling jitter and response time jitter can also be set as requirements so that event handling takes these factors into account. Incorrect handling of jitter could result in a task or data transmission not meeting its timing requirements. In systems without strict timing requirements or in systems where jitter has a lower order of magnitude than the magnitude for which jitter is defined, e.g., delivery latency, jitter can be considered negligible.

As already indicated, tasks and data transmissions do not function stand-alone, but are often part of sequences consisting of multiple processing and data-exchange steps spanning from the moment the real-time system is stimulated by the external environment to a reaction to such a stimulus is delivered. In these circumstances, each step acts as a stimulus for the next in the sequence and requirements can be set for the whole chain rather than for individual event handling, e.g., a delivery deadline for the whole chain. An example of such event chains can be found later in the real-time guarantees in the layered architecture section (Section 4.1.3). However, a characterization of event chains is beyond the scope of this thesis.

Table 4.3 summarizes the extended characterization of tasks and data transmissions.

Table 4.3: Extended characterization of a task τ_i or a data transmission m_i

Characterization	**Notation**
Criticality	X_i
Response time or delivery latency	R_i
Release jitter	$J_i^{release}$
Execution/transmission-time jitter	$J_i^{handling}$
Response-time jitter	$J_i^{response}$

4.1.2 Scheduling

Computing and communication systems have limited resources, which often have to be shared by several applications that require data processing or data transmission, respectively. The role of scheduling is to determine which task or data transmission from those eligible should use the resources at a given time. The resources that scheduling policies deal with are often related to processing or communication channels, but scheduling is sometimes also needed to manage other resources such as memory allocation (Figure 4.6).

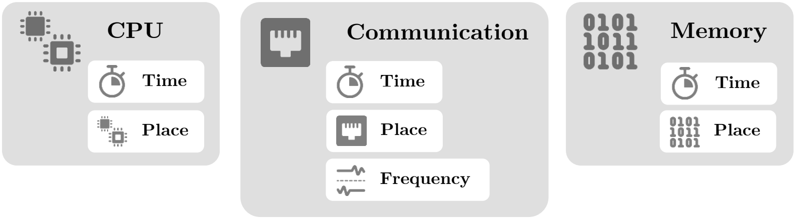


Figure 4.6: Scheduling of computing, communication and memory resources in their various dimensions

Resources are usually scheduled in the temporal dimension, i.e., several tasks or data transmissions can use the resources as long as this use is not simultaneous. However, some resources have other dimensions that can be scheduled in addition to time. This is the case of the transmission channel and the use of diversity schemes, which were explained in detail in the transmission diversity section (Section 2.2.1). The spatial dimension applies, for example, to data that is stored in different memory locations or transmitted via different paths in the network. Similarly, the frequency dimension is effective when the communication system supports data transmissions at different frequencies. In such cases, two or more transmissions could use the resources in parallel as long as the combinations of the different resource dimensions are not identical, e.g., if two transmissions do not take place at the same time, on the same frequency and on the same path.

In some scenarios, there is a mix of tasks or data transfers handling periodic and aperiodic, critical and non-critical events. In such situations, scheduling must first prioritize the handling of critical events over non-critical events while attempting to meet the requirements of periodic and sporadic events. Scheduling could also aim to process unbounded events, yet limiting the interfering effects of such handling over critical tasks or critical data transfers. In distributed systems, processing takes place at different locations that might communicate with each other via communication networks. In such cases, the overall end-to-end deadline requirement is often the goal, rather than meeting the deadline of individual steps in the chain.

The scheduler is the component that applies the scheduling policy or algorithm in the target system. Unless such a policy is straightforward, as in the case of round-robin algorithms where each handling of an event receives an equal share of the resource, it is common for the scheduler to require additional configuration for its operation. The selection of a scheduling policy and its configuration has a significant impact on the performance of the system being scheduled. For example, the policy used for dispatching messages from the MAC layer to the physical layer has a large impact on the time at which these messages arrive at their destination. Therefore, much of the effort in deploying a real-time system goes into engineering and testing schedules that guarantee a high probability of accomplishing timing and perhaps other requirements. In the case of data transmissions, most networks apply simple policies based on first in, first out (FIFO) handling, where the data to be transmitted is sent in the same order in which it arrives in the transmission system. However, some components of the communication stack may be able to control the order in which the data is dispatched by adopting other scheduling policies, as is the case with some of the mechanisms in TSN, for example.

In complex scheduling cases, e.g., with dependencies between tasks and data transmissions, the number of possible combinations that the configuration parameters of the scheduler could adopt, i.e., the search space, makes finding a solution in the so-called solution space a challenging task (Figure 4.7). Optimal assignment algorithms are not applicable to such complex problems and exploring all possibilities by brute force is immeasurable. To solve these cases, heuristic algorithms are often used that do not explore the entire solution space with all configuration options of the scheduler, but only a subset selected from the most promising ones in the near-optimal solution space.

The introduction of a taxonomy of schedulers (Table 4.8) can be based on several criteria. In the context of the offline/static/compile-time vs. online/dynamic/runtime classification, an online scheduler receives this designation if it can be reconfigured during execution in the scheduled system as soon as there are changes to the events to be handled. In contrast, with an offline scheduler, the scheduler configuration must be available before scheduling starts on the target system and does not allow dynamic changes. Online schedulers have the advantage that they can react to dynamic changes in the scheduled workload, but it is often complicated to give guarantees in such a dynamic environment. In contrast, offline schedulers can guarantee the fulfillment of the event handling requirements once a feasible schedule is obtained. However, their adoption is limited by the complexity of solving the scheduling problem, which is often very computationally and memory intensive. A possible compromise solution is to generate multiple offline scheduling configurations and apply them online. However, this option offers limited responsiveness to changes, as only a limited number of schedules can be generated in advance.

Schedulers are also classified as preemptive if they can pause and resume a task or

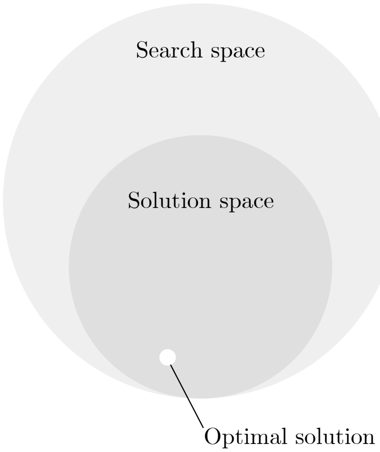


Figure 4.7: Search space, solution space and optimal solution for the scheduler configuration problem

$$\text{Schedulers}\begin{cases}\text{Decision time}\begin{cases}\text{Offline/static/compile-time}\\ \text{Online/dynamic/runtime}\end{cases}\\ \text{Preemption}\begin{cases}\text{Preemptive}\\ \text{Non-preemptive}\end{cases}\\ \text{Handling paradigm}\begin{cases}\text{Time-triggered}\\ \text{Event-driven}\end{cases}\end{cases}$$

Figure 4.8: A taxonomy of schedulers

data exchange during its execution or transmission, respectively. Preemptive schedulers are more flexible in the handling of events, as the handling does not have to continue uninterrupted, but can be paused and resumed depending on the actual needs of the event handling.

A third classification for the schedulers addresses the two paradigms, time-triggered and event-driven, which are usually considered for handling events in the scheduled systems. In time-triggered scheduling, events are handled at scheduled times. If the event to be processed is not present at the scheduled handling time, the handling is skipped. In event-driven scheduling, on the other hand, the next event to be handled is decided either when an event occurs or when the handling of the current event has been completed. Which of the pending events is handled by the event-driven scheduler depends on the specifics of the scheduling algorithm, e.g., the selection of the event with the highest priority or the one with the closest deadline. The distinction between time-triggered and event-driven systems should not mislead the reader. All systems are driven by events, but in the time-triggered case their handling is not only a consequence of the occurrence of the event itself, but the clock initiates such handling, hence the use of different terms. Time-triggered event handling is well suited for periodic events, especially if the generation of events and their handling by the scheduler are synchronized. Aperiodic events often fit better with event-driven schedulers, as they offer the possibility to react to events as soon as they occur, depending on the specifics of the scheduling algorithm and at the cost of displacing the handling of other events. Combinations of different types of schedulers under the same system, e.g., by applying a different scheduling policy per processing core, are often helpful to get the best out of each of the options.

Schedulability analysis is used to evaluate whether a set of tasks and data transmissions are able to fulfill their timing requirements under a given combination of scheduling algorithm and scheduler configuration. Schedulability analysis is readily available for the most common even-driven scheduling algorithms such as fixed priority (FP) or earliest-deadline first (EDF), but only in simple scenarios without dependencies between tasks or data transmissions. For more complex cases, analyzing schedulability is anything but simple. In time-triggered scheduling, the scheduler configuration allocates time slots for processing events, and such assignment is often made already taking into account that processing is guaranteed enough time to complete and time requirements are met. Hence, it is not necessary to perform a schedulability analysis as schedulability is guaranteed once a feasible time-triggered schedule has been created. The schedulability analysis is traditionally developed with a focus on tasks and processors. However, schedulability analysis can also be outlined similarly for data transmissions. In this case, data transmissions are treated as tasks and the transmission time is comparable to the execution time of the task. However, these strategies must take into account that, unlike tasks, the scheduler in a communication system is usually not able to preempt data transmissions.

Event-driven scheduling

FP (Figure 4.9) is a static, event-driven scheduling policy for task processing in the most widely used operating systems, e.g., Linux and Windows. With FP, events are handled on the basis of a pre-assigned priority. In combination with preemption, the processing of a running task or data transmission is paused as soon as a high-priority event arrives in the system. The preferential handling of high-priority events makes them get lower response times, but can result in low-priority events being neglected. FP scheduling is also very sensitive to fluctuations in activation and execution time. Minor changes in activation or execution time could have a strong impact on the outcome of the overall event handling, making determinism guarantees challenging to obtain. For FP, there are several algorithms for priority assignment, e.g., rate monotonic or deadline monotonic, which provide optimal results under certain conditions. In combination with the analysis of response times, it is possible to determine whether tasks or data transmission can meet their deadlines. The analysis is based on checking the deadlines in the worst case, which happens when all events occur at the same time. The schedulability of an event handling is influenced by the handling of higher-priority events, the own event handling time and the blocking time caused by the handling of lower priority events. The latter can occur for various reasons, e.g. due to sections of program code that cannot be preempted or due to synchronization points between different event handlers for accessing shared resources.

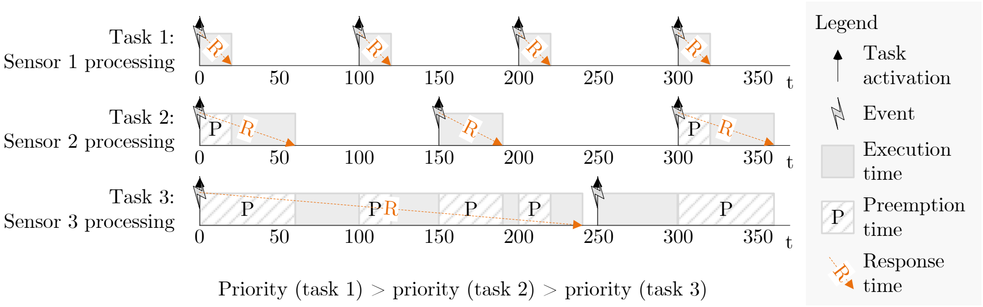


Figure 4.9: Example of FP scheduling

Together with FP, EDF (Figure 4.10) completes the pair of the most common event-driven scheduling policies. According to the EDF scheduling policy, an event is selected for handling when it has the closest deadline in the system. EDF dynamically checks which event should be processed each time a new event is available in the system or when the handling of an event has just been completed. EDF is usually combined with preemption to ensure that the current event handling is paused when an event with a more urgent deadline is present in the system. The EDF scheduling policy analysis is able to calculate the response time for event handling. The worst-case analysis for EDF assumes that all events are activated simultaneously and experience their maximum activation jitter, so that they are activated at the closest instant to their deadlines.

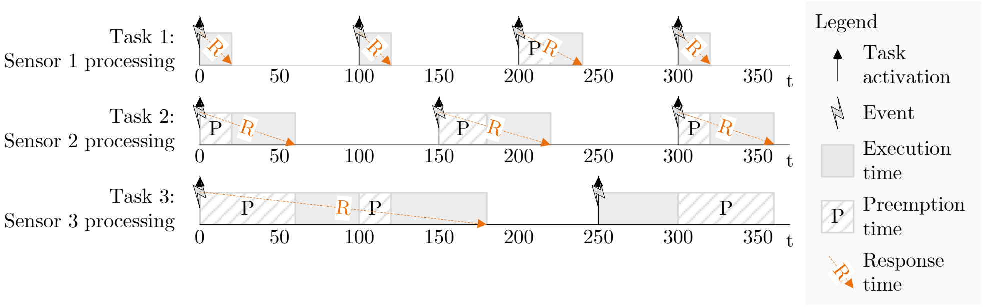


Figure 4.10: Example of EDF scheduling with task deadlines equal to their periods

The analyzes for FP and EDF often assume a periodic event model with certain limitations, such as no dependencies between tasks and data transmissions. More complex scenarios therefore require further and more complex analyzes, which are often not readily available.

For sporadic events, guarantees can be provided by analyzing them like periodic events by setting the period to the maximum rate or the MIT.

In the case of unbounded events, full timing guarantees cannot be given as the number of events is unlimited by definition. However, an average response time can be pursued while trying to limit the impact on the other tasks or data transmissions in the system. Unbounded events could be serviced with high priority on an FP scheduler if they are critical, or use a low priority if they are not critical and claim the background time left over from higher priority tasks or data transmissions. The first option provides the best possible response time for unbounded events, but also impacts the determinism of periodic and sporadic tasks or data transmissions. The option to use background time can cause many unbounded events to miss their deadlines, but without interfering with the handling of periodic and sporadic events. Unbounded events can also be handled periodically by using a polling server. The response time then depends on the period in which the polling takes place, with a shorter period reducing the response time but impairing the schedulability of other tasks or data transmissions. A frequent handling may not be sufficient to serve enough unbounded events. A flexible approach is offered by sporadic servers. Sporadic servers can take different approaches, but very often they are characterized by their capacity and replenishment period. The capacity represents the time available to handle events and decreases each time events are handled. When the capacity drops to zero, no more events are handled until the capacity is replenished after the replenishment period. It is also possible to use a purely EDF-based server with the so-called total bandwidth server, where aperiodic tasks or data transmissions are given a deadline. Sporadic servers offer reduced response times when the capacity is still available and protect the system from unbounded incoming events when the capacity is zero. Polling, sporadic and bandwidth servers can be introduced as part of the periodic task or data transmission model, with the polling and replenishment period acting as the period of the event and the capacity taken as the handling time. The time that the server has for processing the events can also be defined according to FP or EDF policies. In this case, the server is given a priority or a deadline, respectively to the FP or EDF policy, and is treated like any other task in the system. As it can be seen, there are many methods for handling events, and the possibilities become even greater when different methods are combined in multiple ways. This makes it difficult to know how such combinations would affect, for example, the predictability of events and the response time. Figure 4.11 lists the event-driven schedulers covered in this section.

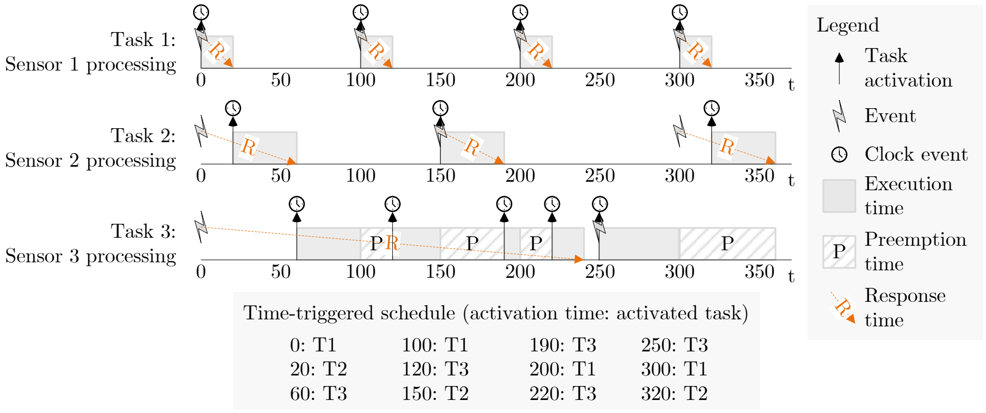
$$\text{Event-driven schedulers}\begin{cases}\text{FP}\\ \text{EDF}\\ \text{Polling server}\\ \text{Sporadic server}\\ \text{Total bandwidth server}\end{cases}$$

Figure 4.11: Taxonomy of event-driven schedulers

Furthermore, analyzing the response time for a distributed system in which there are chains of processing and communication steps is also anything but simple. This is because the activation of the events in the chain, except for the first event, depends on the response time of the handling of the previous event in the chain. Since the response time can vary between different iterations, this directly leads to jitter in the activation time of subsequent event handlers in the chain. An iterative approach can calculate the response time of events and translate it into the activation time for the following events in the chain until a solution converges and the response time for the entire chain is obtained. However, the solution obtained with this method is quite pessimistic. Other techniques improve this result by using the offsets to indicate when events are activated and analyzing each computing and communication segment independently. However, such details are beyond the scope of this thesis.

Time-triggered scheduling

With time-triggered scheduling (Figure 4.12), tasks and data transmissions are handled at pre-defined instants according to a schedule. Since the time at which tasks and data transmissions are handled can be selected by the scheduler, it is possible to adjust their placement in the timeline to accommodate their timing requirements, including modeling all types of jitter. If tasks attempt to exceed their allocated budget, they can be preempted so that the time allocated to other tasks is not taken up. The pre-assigned budget is a major advantage of time-triggered scheduling, as it allows different applications to be easily integrated with their tasks and data transmissions, as the temporal isolation in the use of resources causes them not interfere with each other. In the case of communications, it is often not possible to preempt a transmission that exceeds its budget. However, transmissions generally do not exhibit fluctuations in their transmission times, and if these are to be expected, e.g., due to a rate control algorithm, the data delivery time should take them into account.



Time-triggered schedule (activation time: activated task)

0: T1	100: T1	190: T3	250: T3
20: T2	120: T3	200: T1	300: T1
60: T3	150: T2	220: T3	320: T2

Figure 4.12: Example of time-triggered scheduling

Time-triggered scheduling comes with a number of limitations. One of them is that the systems to be scheduled and the originators of events must be synchronized in order to achieve the best response time. In addition, there is limited flexibility in responding to unplanned events. For example, if the scheduler does not assign time to unbounded events, they will not be handled. Another downside of time-triggered scheduling is that the worst possible handling time is often taken into account to create the schedules. Such an allocated handling time is often not fully utilized and cannot always be reallocated to handle other events, as would be the case with FP and EDF. There are many ways to make the time-triggered schedule more flexible, e.g., by allocating handling time to groups of tasks that are scheduled within the budgeted time according to an event-driven scheduling policy.

Given the set of tasks and their communication requirements, the time-triggered scheduler tries to find an allocation of computational or communication resources, respectively, in the time domain. Other domains, such as space or frequency, may also need to be considered. The search for a solution to the problem of time-triggered scheduling is NP-complete in terms of computational complexity. All possible solutions must be explored to find the optimal solution.

The formulation of the time-triggered scheduling problem, i.e., the processing resources, tasks, network topology and data exchanges, in the form of first-order logic (FOL) constraints provides a mathematical way to represent the requirements to the scheduling problem. To obtain a solution, such a formulation can be handled over to meta-heuristic algorithms, or scheduling tools based on satisfiability modulo theories (SMT) or integer linear programming (ILP), among others. The works of [11] and [38] prove the feasibility of these methods for time-triggered scheduling. Nevertheless, the details of how to find a valid solution for the FOL problem formulation using the mentioned methods are beyond the scope of this thesis.

4.1.3 Real-time guarantees in the layered architecture

The transmission of data between a sender and a receiver involves several components that take over the responsibilities from the different layers in the communications architecture. The presence of multiple components and steps can complicate the fulfillment of timing requirements. Figure 4.13 shows an example of common components involved in data communication between two general-purpose systems where intermediate communication nodes, e.g., switches, are present. Both systems have sending and receiving applications taking over the application layer responsibilities. They also have a network interface card (NIC), i.e., the network hardware provides the end systems with communication capabilities. The NIC assumes the responsibilities of the physical layer and partly those of the data-link layer. Finally, an operating system on both sides handles the responsibilities between the data-link layer and the application layer.

To ensure timely data exchange between the applications at both ends of a communication system, each of the components involved must fulfill its duty within certain time boundaries. These limits may be tighter depending on the jitter requirements. However, even if some of the components can be precisely scheduled in certain situations, e.g., the data dispatching by the switch nodes or the applications that generate the data, there are others that might escape such tight control, e.g., the forwarding of data from the operating system to the NIC. In such cases where a tight control is not possible, defining a large enough time budget in which these duties are most likely to be completed may be sufficient to guarantee that the subsequent steps in the chain receive the data in a timely manner. Figure 4.14 shows an example of a timeline for the components of the communication stack, which shows that some components can be scheduled precisely. In contrast, other components require some slack in order to increase the likelihood that they will fulfill their duties.

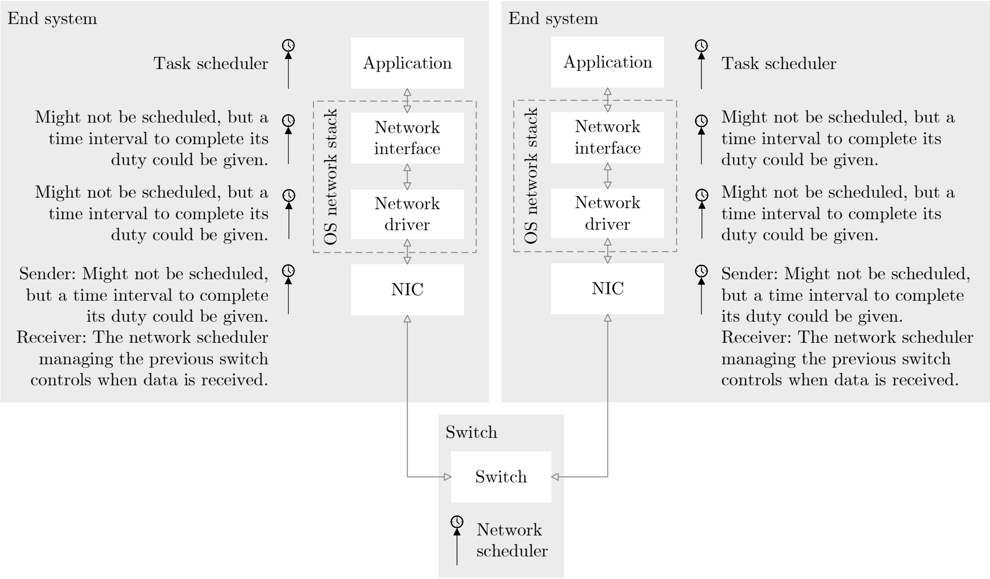

End system
Task scheduler
Application
Might not be scheduled, but a time interval to complete its duty could be given.
OS network stack
Network interface
Might not be scheduled, but a time interval to complete its duty could be given.
Network driver
Sender: Might not be scheduled, but a time interval to complete its duty could be given.
Receiver: The network scheduler managing the previous switch controls when data is received.
NIC
End system
Application
Task scheduler
Network interface
Might not be scheduled, but a time interval to complete its duty could be given.
Network driver
Might not be scheduled, but a time interval to complete its duty could be given.
NIC
Sender: Might not be scheduled, but a time interval to complete its duty could be given.
Receiver: The network scheduler managing the previous switch controls when data is received.
Switch
Switch
Network scheduler

Figure 4.13: Example of modules involved in the communication between two end systems and description of how the modules are scheduled.

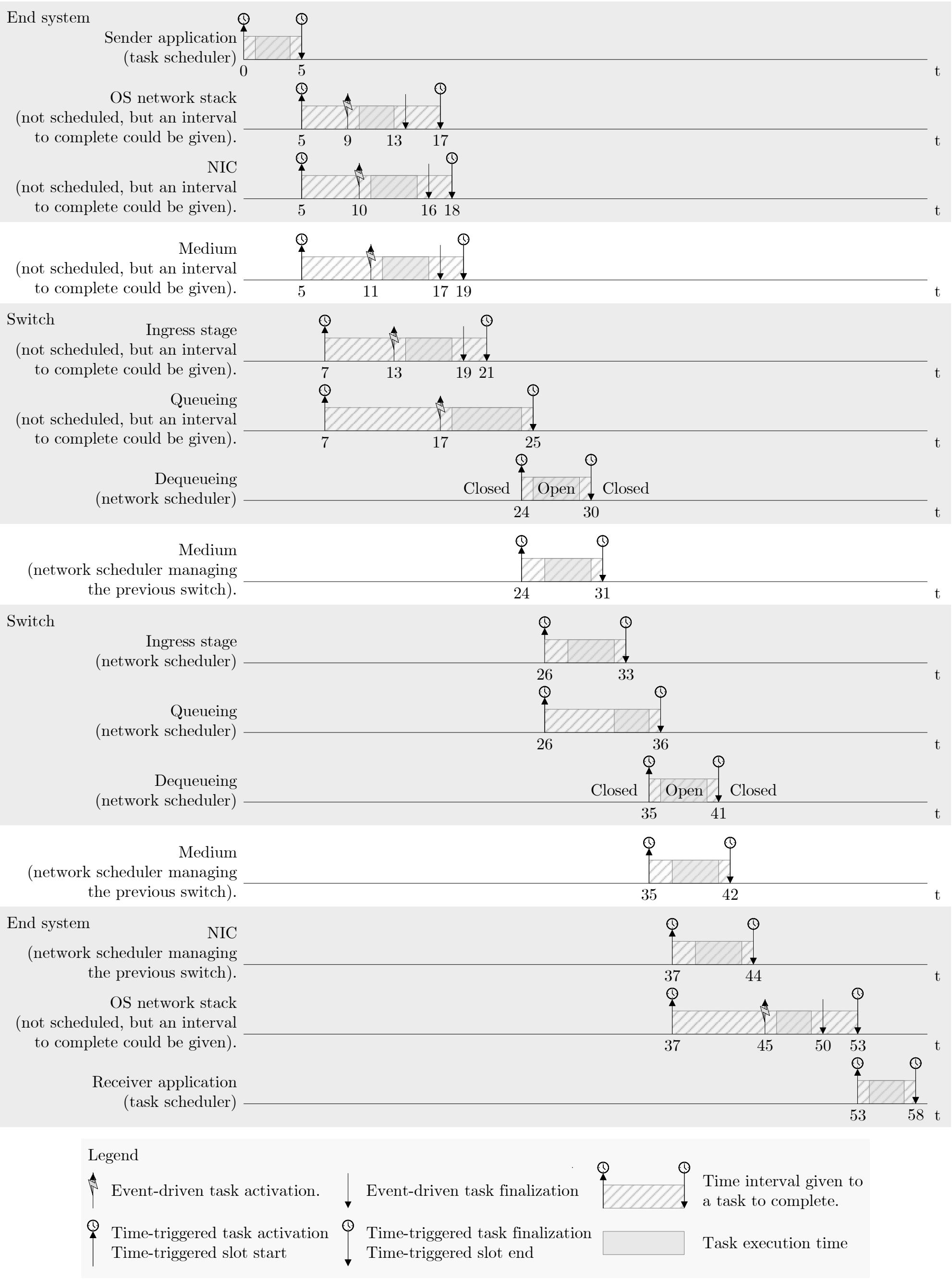

End system
Sender application
(task scheduler)
0
5
t
OS network stack
(not scheduled, but an interval
to complete could be given).
5
9
13
17
t
NIC
(not scheduled, but an interval
to complete could be given).
5
10
16
18
t
Medium
(not scheduled, but an interval
to complete could be given).
5
11
17
19
t
Switch
Ingress stage
(not scheduled, but an interval
to complete could be given).
7
13
19
21
t
Queueing
(not scheduled, but an interval
to complete could be given).
7
17
25
t
Dequeueing
(network scheduler)
Closed
Open
Closed
24
30
t
Medium
(network scheduler managing
the previous switch).
24
31
t
Switch
Ingress stage
(network scheduler)
26
33
t
Queueing
(network scheduler)
26
36
t
Dequeueing
(network scheduler)
Closed
Open
Closed
35
41
t
Medium
(network scheduler managing
the previous switch).
35
42
t
End system
NIC
(network scheduler managing
the previous switch).
37
44
t
OS network stack
(not scheduled, but an interval
to complete could be given).
37
45
50
53
t
Receiver application
(task scheduler)
53
58
t
Legend
Event-driven task activation.
Event-driven task finalization
Time interval given to
a task to complete.
Time-triggered task activation
Time-triggered slot start
Time-triggered task finalization
Time-triggered slot end
Task execution time

Figure 4.14: Example of the communication steps between two end systems and description of how the steps are scheduled.

4.2 Dependable systems

Dependability is a property of systems that can be trusted upon to function as expected. The expected behavior is referred to as the service of the system and is described by its requirements. Therefore, a dependable system is such that it adheres to its requirements.

Dependability cannot be defined without referring to the attributes that make the system dependable. Unfortunately, these attributes can be compromised in different ways. Thus, a system is only dependable if it can cope with the threats that can be expected for each of these attributes, which vary depending on the system and the scenario. Both the attributes and the threats are described next.

4.2.1 Dependability attributes

The attributes that make a system dependable vary according to the definitions of the different authors, often due to the historical evolution of the systems and what is of more relevance for the scenarios in which the systems are deployed. The definition selected in this thesis [9] comprises the attributes of reliability, availability, safety, maintainability and integrity (Figure 4.15). A description of the attributes for the specific case of data communications was given as part of the dependability requirements in Section 3.3. According to the same definition [9], security is considered a system property that, like dependability, comprises several attributes, namely availability, confidentiality and integrity. For a system to be secure, the service must therefore be available and the data must be treated confidentially and with integrity. Other definitions of dependability [10] also include performability or testability. The former is the ability of the system to provide its service within certain limits. This aspect is covered in this thesis by the overhead and efficiency requirement. The latter is the ability of the system to be tested, an attribute that falls outside the scope of this thesis.

$$\text{Attributes} \begin{cases} \text{Reliability} \\ \text{Availability} \\ \text{Safety} \\ \text{Maintainability} \\ \text{Integrity} \end{cases}$$

Figure 4.15: Dependability attributes [9]

4.2.2 Threats to dependability

A system might deviate from its dependability goals for a variety of reasons. It is therefore important to recognize the causing factors and take countermeasures. Such deviations begin with a fault, a term that refers to a defect in the system. A fault can either manifest itself in the form of an error or stay dormant until it eventually occurs, if it occurs at all. An error causes the system to deviate from the way the service shall be performed, but the service might still be provided correctly. Only if the service is not provided correctly there is a failure. Figure 4.16 depicts the relationship between fault, error and failure.

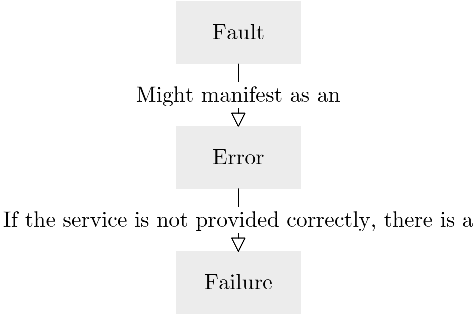


Figure 4.16: Fault, error and failure

Since failures are deviations from the expected service, a thoughtful definition of the system requirements is crucial to define what is and what is not correct service. The actual definition of requirements can be a source of potential failures if the process does not correctly describe what the system is expected to do.

The report in [9] describes faults on the basis of eight aspects (Figure 4.17). The aspects that are most important for this thesis are mentioned next. According to the system boundaries, faults have an internal or external origin. For example, a violation of the transmissions schedule that causes them to occur simultaneously in a real-time communication system is likely to have an internal cause. In contrast, a transmission affected by interference has an external reason behind it. According to the objective, faults are classified into malicious and non-malicious. The use of jamming signals to create interference is an example of a malicious objective that requires security reinforcement. In contrast, interference from other devices in the area trying to transmit data genuinely is an example of the opposite. Related to the objective aspect, the intent distinguishes between deliberate and non-deliberate faults. Deliberate faults result of an informed decision that may not have a malicious objective behind it, but other reasons such as cost or complexity reduction. For example, systems with hard real-time requirements often rely on non-real-time operating systems or communication networks to accomplish their duties because these systems are readily available and have many features that make them convenient to use. In such circumstances, it might be advisable to gain sufficient confidence in the system performing according to requirements, maybe through validation based on continuous testing or runtime monitoring. One last aspect of faults that is relevant to this thesis is their persistence. The so-called transient faults can occur and disappear after some time, while permanent faults do not disappear after they occur. For instance, wireless interference is usually time-dependent. Thus, the faults caused by interference are usually transient. A broken wired connection, on the other hand, is considered a permanent fault because it does not disappear on its own unless the cable is repaired.

Once a fault manifests as an error, the error is either detected or stays latent. If the error is detected, it can be notified to the user via some message or signal. For

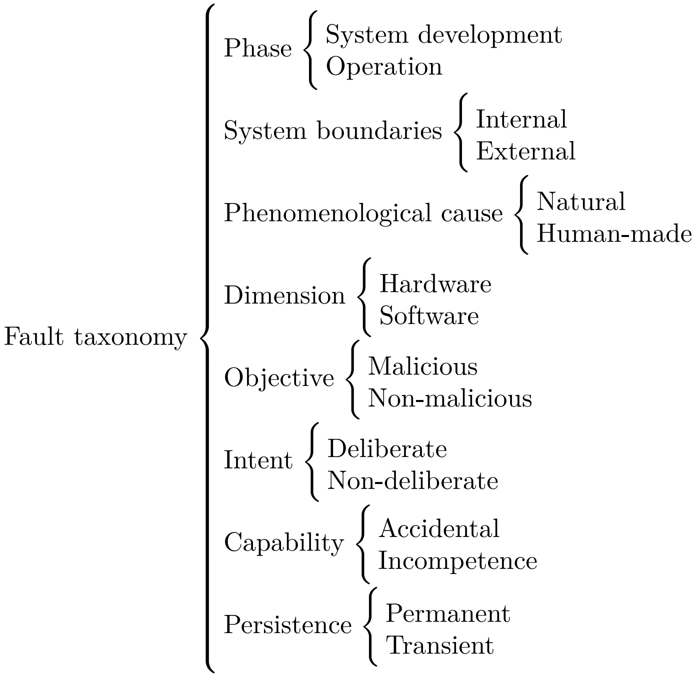


Figure 4.17: Fault taxonomy [9]

example, a fault caused by interference and affecting a data transmission could be detected by the sender because it has not received an acknowledgment of receipt from the receiver. The sender could report an error or the fault could go unnoticed if there is no such feedback mechanism.

The work in [9] also describes a taxonomy of failures that includes aspects regarding domain, detectability, consistency and consequences (Figure 4.18).

The failure domain refers to incorrect content and timing aspects of an output generated by the system. For example, if the system is supposed to output a temperature value in kelvin units and returns a negative value, which is not possible in kelvin units, this is a content failure. For systems with timing requirements, the time at which the output is generated is as essential as the content; otherwise, early or late timing failures will occur. Finally, some failures are a combination of both content and timing characteristics, with halt failures happening when the system fails to produce a result, and erratic failures when both the content and timing are incorrect.

According their detectability, failures are classified into signaled and non-signaled failures, depending on whether the system notices them. The use of acknowledgments (ACKs) and timeouts is an example of a mechanism used in communication systems to provide feedback on the success of a transmission. However, such failure detection mechanisms can also fail, leading to four possible cases that are part of the so-called confusion matrix (Table 4.4). When using ACKs, for example, the reception of a transmission must be followed by an ACK sent by the data receiver to the sender

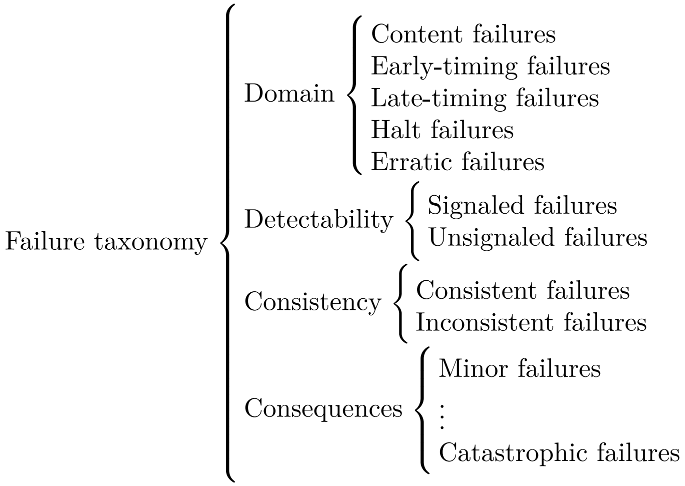


Figure 4.18: Failure taxonomy [9]

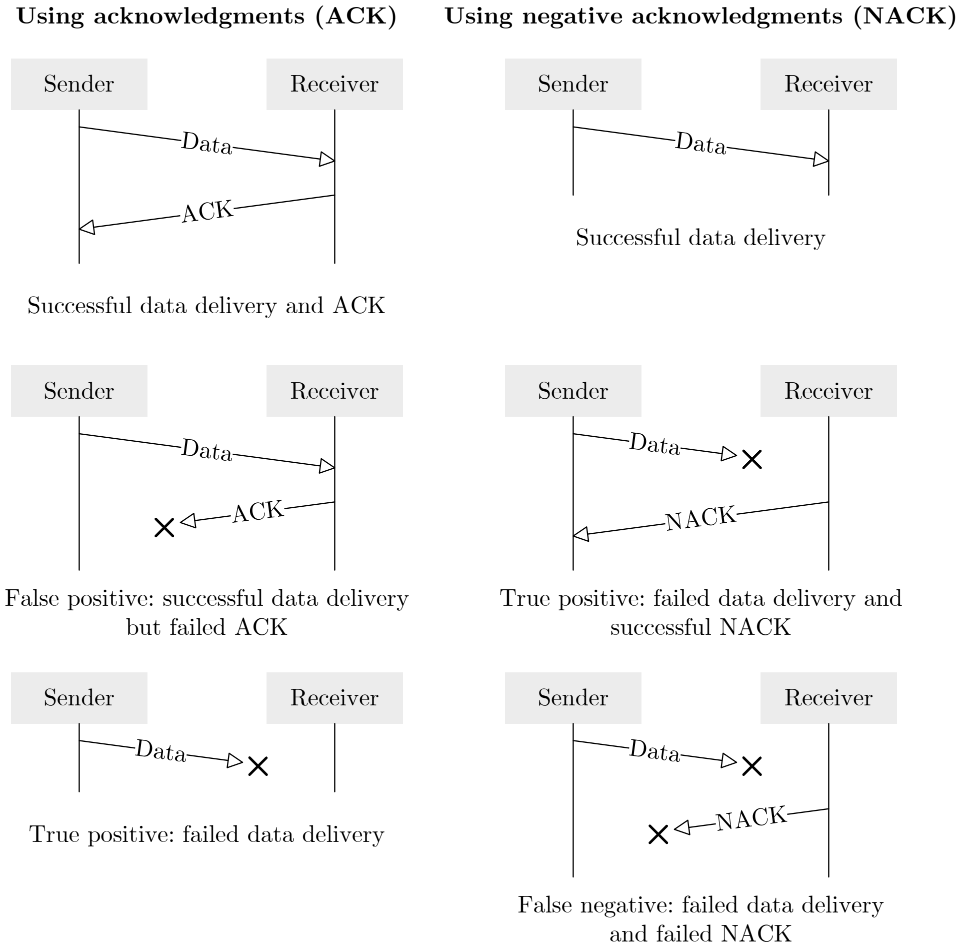
(Figure 4.19). Such an ACK might not be received by the data sender because it has suffered a fault, e.g., due to interference. In this case, the sender assumes that the data transmission has failed, although the data has arrived at the destination, resulting in a false positive. A false negative, on the other hand, occurs when negative acknowledgments (NACK) are used, which the potential recipient always sends when it expects to receive data but has not received it. The non-receipt of a NACK could be interpreted by the sender as confirmation of successful delivery. However, the NACK could have been interfered.

Table 4.4: Failure confusion matrix

		Actual	
		Failure occurs	**Failure does not occur**
Predicted	**Failure detected**	True positive	False positive
	Failure not detected	False negative	True negative

Failure consistency refers to how a group of users perceives failures. With consistent failures, all users perceive the failure in the same way, whereas with inconsistent failures, different perceptions are gathered. For example, broadcast transmissions are expected to be received by all nodes in the so-called broadcast domain. If, for example, some transmissions are faulty due to interference, the data may not arrive at some of the expected receivers while others received it. Those who receive the data will then assume that there was no failure, while those who do not receive the data will do so.

The failure consequences are used to indicate the severity of their occurrence. The consequences refer to the criticality levels described as part of the criticality requirements in Section 3.1. The definition of the consequences or criticality levels



Using acknowledgments (ACK)

Using negative acknowledgments (NACK)

Successful data delivery and ACK

Successful data delivery

False positive: successful data delivery but failed ACK

True positive: failed data delivery and successful NACK

True positive: failed data delivery

False negative: failed data delivery and failed NACK

Figure 4.19: Examples of failed and non-failed data transmissions using ACKs and NACKs.

depends heavily on the use case, and they can range from minor to catastrophic. In avionics, for example, the DO-178B standard [39] defines five levels of criticality, E to A, which are categorized as none, minor, major, hazardous and catastrophic, depending on their consequences. An extract from this standard is given to convey the idea of the limits of the specified criticality range. None is described as “no effect on operational capabilities or safety margin” and catastrophic as “safe flight and landing prevented, usually with loss of aircraft”. Each of the DO-178B categories has an associated maximum probability of occurrence that compliant systems shall observe.

4.2.3 Dealing with threats

A dependable system is categorized as either fail-safe or fail-operational based on its response to the presence of faults. In fail-safe systems, a failure results in the system reaching a safe state in which no harm is caused to the user or the environment, but no service is provided. This is the case, for example, when an industrial conveyor belt is stopped as soon as a failure is detected that causes the managed products to pile up if it remains in operation. In other cases, minimum functionality must be provided to avoid the consequences of a catastrophic failure. Such systems are fail-operational, with examples including planes or nuclear reactors. The service in a fail-operational scenario is a subset of the regular service provided by the system, i.e., a degraded functionality and the minimum required to remain safe when some requirements are not met.

Dependability means

Considering that the consequences of some failures are unacceptable, especially when they are closer to the catastrophic categorization, handling faults becomes necessary to reduce the probability of failure to a level that is adequate for the given scenario. The work in [9] describes four means of tackling with faults in such a way that dependability is observed: fault prevention, fault tolerance, fault removal and fault forecasting (Figure 4.20).

$$\text{Means}\begin{cases}\text{Fault prevention}\\ \text{Fault removal}\\ \text{Fault tolerance}\\ \text{Fault forecasting}\end{cases}$$

Figure 4.20: Dependability means [9]

In fault prevention, efforts are made to reduce the occurrence of faults. For example, a coding technique may be used in a communication system to reduce the effects of interference and decrease the likelihood that data different to the one sent is delivered, causing the system to suffer from a fault. Unfortunately, there are limits to fault prevention, as the components of a system cannot be improved indefinitely and there is always a certain probability that they will fail. Such limits can be theoretical or determined in practice. For instance, Shannon's theorem sets an upper limit on the throughput of a communication system. If such a bound did not exist, an infinite amount of information could be transmitted within a time window, so that the presence of faults would be overcome by retransmitting the data at no cost, which is unrealistic. Some other limitations are based on the trade-offs that must be made in the requirements gathering process. For example, costs in terms of time and money often set the limits of fault-prevention efforts.

Fault removal takes an existing system and verifies whether it complies with its specifications. Based on the results of the verification, an attempt is then made to correct any deviations from the specification that lead to faults.

In the case of fault tolerance, the system is designed in such a way that it can deal with faults. This means that the system can continue to provide its service even in the event of a fault. A fault-tolerant system must detect the fault through monitoring before it reacts to it. The reaction could be based on removing the fault by bringing the system to a previous or later state where the fault is not present. Alternatively, the fault could be compensated for by sufficient redundancy, a case that is explained in the next section.

Finally, fault forecasting focuses on studying the faults in the system, how often they occur, under what conditions and with what consequences. Performing tests in which faults are injected into the system is a common method to check the reaction of a system to faults.

Improving dependability: redundancy as a key mechanism

Redundancy is often used to enable fault tolerance and avoid the problem of having a single component that fails, i.e., a single point of failure. The price to pay for redundancy mechanisms is the additional resources required to replicate components. For example, communication networks may have two or more redundant paths for transmitting data from a sender to a receiver. If one of the paths is faulty, e.g., due to a broken cable, the system can continue to provide its service by using the additional paths. According to [10], redundancy can be achieved by replicating hardware and software or using individual resources, but at different times, i.e., time redundancy (Figure 4.21).

$$\text{Redundancy}\begin{cases}\text{Hardware}\\\text{Software}\\\text{Information}\\\text{Time}\end{cases}$$

Figure 4.21: Redundancy taxonomy [10]

With hardware redundancy, the replicated hardware components can be used according to an active or passive redundancy approach (Figure 4.22). With passive redundancy, also known as hot standby, all replicated components work in parallel and are stimulated by the same inputs. The replicated components use the inputs to generate the outputs. The outputs then feed a voter module that selects the most repeated output. If none of the outputs turns out to be the most frequently repeated, no output is selected, but a failure is detected. The voter itself is another component, but it can be assumed to be simpler than the replicated components and should not be prone to suffer from faults. Nevertheless, architectures with multiple voters can be used to try to overcome the problem of a single point of failure at the voter. It should be noted that if the replicated components are identical, some faults may occur in all components simultaneously. Designing hardware with the same functionality, but independently, should help to overcome this issue. With active redundancy, also known as cold standby, one component is active at a time and the remaining replicated components wait for a fault in the primary component to be activated. Active redundancy allows for a lower resource consumption and lower component wear. However, detecting the fault could become more complicated than the simple voter mechanism. Also, putting the standby components into operation is likely to delay the reaction to a fault compared to passive redundancy, which is its biggest drawback. The prolonged reaction time could mean that the system suffers from a transient failure until the standby components start operating. Nevertheless, there can be architectures in which passive and active redundancy mechanisms are combined. Passive and active redundancy are often associated with hardware redundancy, but both can also be applied to the other types of redundancy described below.

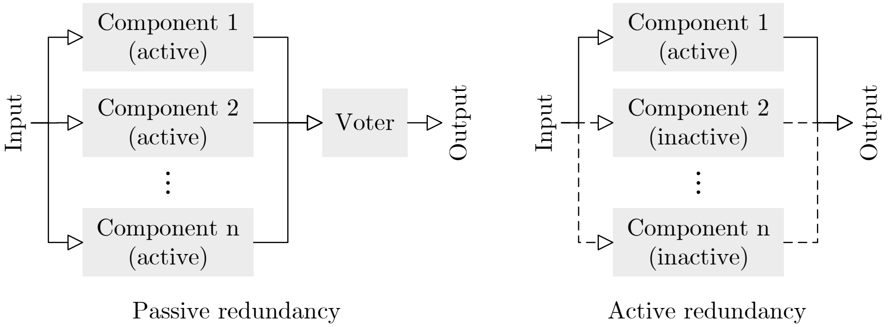


Figure 4.22: Passive and active redundancy

Software redundancy applies fault tolerance at the software level. Some of the software redundancy mechanisms consist of introducing checks on the consistency of information or tests to evaluate the capabilities of the software. Similar to the development of independent hardware components to eliminate the common cause of failures, N-version programming proposes that N programs with identical functionality are developed by independent teams to reduce the likelihood of making the same mistakes in all programs.

In information redundancy, redundant data is added to facilitate the detection and correction of faults. The use of coding in digital communications is an example of information redundancy. This involves adding extra bits to data transmissions to increase the likelihood of the data being successfully received.

In a time redundancy mechanism, resources are not replicated, but their use is spread over time. A time redundancy mechanism can be applied only after checking the presence of faults, i.e., active redundancy, or by default without any check, i.e., passive redundancy. Time redundancy works better in the case of transient faults, e.g., when a temporal interference affects data transmissions. However, if the reason for suffering faults is permanent, trying to perform the same action at different times will likely lead to the same undesirable results. Time redundancy also has limitations in terms of the number of possible repetitions of a potentially-faulty action, which is particularly relevant for meeting deadlines in real-time systems. In these cases, scheduling the redundant steps is often the best way to ensure that timing constraints are met while applying fault tolerance.

The concept of fault coverage is often used to assess the extent to which redundancy mechanisms help to counteract faults. Fault coverage refers to the ability of the system to detect its faults. A common method for determining coverage is experimental fault injection, where faults are introduced into the system to determine how the system reacts to them.

4.2.4 Dependability analysis: the case of reliability

Estimating reliability is crucial for systems with real-time and criticality requirements. Reliability can be evaluated experimentally by running the target system for a period of time and extracting the failures that occurred during this period, i.e., the failure rate. This estimation is possible because reliability and failure probability are inverse functions. Failure rates are often modeled constant but can also be described using functions such as Weibull, which reflects the usually higher failure rates at the beginning and end of a component's life. However, such estimates have their limitations as tests for certain systems are sometimes not easy to set up and especially because a lot of time needs to be invested to have some confidence in the results. Hence, a common approach is to use models that resemble the behavior of the system. These models are based on calculating the reliability of the individual components in order to be realistic. Unfortunately, such an estimate is often complicated to obtain. Even if such precise estimates are lacking, a reliability estimate can still help to give an idea of performance when, for example, comparing several systems or systems with different design alternatives.

The use of reliability block diagrams (RBD) is a common option for reliability analysis of systems where components are connected in series or parallel. Series connections are used to model systems where each of the components depends on the preceding ones to function. In contrast, only one of the components connected in parallel is required for the group of components to work. Each of these combined components counts with a mathematical expression to calculate the reliability of the RBD model. Arbitrary block diagrams extend RBD with more complex interactions, e.g., it is possible to model active and passive redundancy. Unfortunately, building the RBD or the arbitrary block diagram for complex systems is not straightforward. Other mechanisms such as Markov chains or generalized stochastic Petri nets are more powerful than RBD, as they enable the modeling of more complex scenarios.

Chapter 5

Local and personal area network technologies

Reviewing existing communication technologies and evaluating the extent to which they satisfy the communication requirements is crucial for the introduction of new solutions such as those presented in this thesis. The most important features of LAN and PAN technologies are presented below. The technologies are divided into IT and OT as well as wired and wireless options. LANs have traditionally been associated with IT networks deployed in home and office environments. However, OT networks could also be classified under the LAN term according to the area criteria. Although PAN technologies could fit equally well in the IT and OT split, the reviewed PAN examples are mainly selected due to their applications in the OT field. A comparison of the technology features can be found at the end of the chapter together with a summary of the degree to which the LAN and PAN technologies meet the selected communication requirements.

5.1 Information technology

The introduction of networks for IT use was fueled in the 1990s and 2000s by the increasing number of computer systems, first in the office environment and later in the home environment. The introduction was primarily founded on Ethernet and IEEE 802.11. IT technologies have more relaxed dependability and real-time requirements than OT, with high throughput and lower installation costs often cited as prominent features that caused their wide adoption.

Wired networks were deployed first in IT due to the rather stationary computer systems at that time, which were far from being portable. Ethernet is considered to be the undoubted leader in the wired IT area. With the proliferation of consumer electronics, and smaller and portable computer systems, IEEE 802.11 wireless technology was introduced until it later became the leading option for networking in home environments and a widely adopted alternative in office environments.

5.1.1 Ethernet

IEEE 802.3 “Ethernet” [12] has been a success story since it first appeared in the 1980s, when it was gradually adopted for IT applications. Today, it is a mature technology that results very familiar to computer users and as a consequence of its popularity became quite affordable. The standard defines several options for the physical layer, with the most widespread versions capable of transmitting at 100 Mbps, 1000 Mbps and 10 000 Mbps in full-duplex mode, and transferring data sizes up to 1500 bytes of physical-layer payload. The connection between nodes has also evolved from the early bus-based topologies, where each node listened to every other ongoing transmission in the network, to switched Ethernet, which is currently the most common alternative. Switched Ethernet enables ring, star and tree topologies. Each node has one or more ports, each of which forwards data to a different set of destination nodes based on the destination address, i.e., the MAC address. The topology freedom and the available data rates benefit the scalability of the technology.

At each port there is a queue in which the data waits for ongoing transmissions to finish. The transmissions are carried out according to a MAC protocol based on CSMA/CD. However, considering the switched-Ethernet topology and the full-duplex links, only one sender is allowed to transmit in each direction on a link, effectively avoiding collisions on the transmission medium. The simplicity of the MAC algorithm, which does not require complex configuration, enables applications to easily add or remove data transmissions, making Ethernet a very scalable and adaptable technology. Unfortunately, the protocol cannot provide timing guarantees as data from different sources interleaves at the switches, which can lead to delays and data loss. Such a scenario occurs when the incoming data overflows the capacity of the switch to redirect the data or the incoming data exceeds the capacity of the outgoing links that cannot keep up with the pace. Under such circumstances, the receiving node can notify the sending node to halt the dataflow until it can receive more data by using the so-called pause frame. The MAC protocol on Ethernet does not offer any support for distinguishing data with different criticality, so transmissions of high criticality can get affected by low criticality transmissions.

Ethernet also lacks redundancy, acknowledgment of frames or clock synchronization mechanisms, which are left as the responsibility of upper-layer protocols. Nevertheless, Ethernet is an enabler for the TCP/IP stack. It also comes with a whole set of protocols that support the dynamic connection of Ethernet nodes. Among these protocols, the link-state routing protocols such as the Spanning Tree Protocol (STP; IEEE 802.1Q) were originally developed to avoid loops in Ethernet networks and to enable alternative routes after topology changes. The protocol exchanges the views of the individual routers via the network topology until the entire network converges. With STP, the protocol goes through several phases, which can take up to 50 s to complete. Faster convergence can be achieved with the Rapid Spanning Tree Protocol (RSTP; IEEE 802.1Q), but still in the order of seconds, a time that is too long for a large proportion of real-time use cases. Spanning tree protocols have evolved and the current state of the art in Ethernet is Shortest Path Bridging (SPB; IEEE 802.1aq [26]), a protocol described later in the TSN section.

5.1.2 IEEE 802.11

802.11 “Wi-Fi” [13] is the IEEE standard for wireless local area networks that applies to the physical and data-link layers. The introduction of this technology in offices, homes and public spaces is ubiquitous. Its overwhelming success is due in part to the fact that it is often seen as the wireless counterpart to Ethernet, with an easy connectivity between these two technologies. This connectivity is provided by affordable devices based on IEEE 802.1 bridging protocols and IEEE 802.2 logical link control (LLC), which abstracts the MAC, regardless of whether the underlying protocol is Ethernet or IEEE 802.11. IEEE 802.11 is a technology intended for best-effort service. Its offer for timely data delivery is not widely adopted and its performance rapidly degrades with the number of users and data transmissions [40][41].

The standard refers to the network nodes as stations (STAs). STAs can take on the additional role of access points (APs), i.e., AP/STA, if they act as intermediate nodes and as gateways to Ethernet networks. The standard defines three ways of connecting STAs to each other. First, directly between STAs via ad-hoc connections conforming an independent basic service set (IBSS). Secondly, via an AP, which is the central node in a start topology and creates an infrastructure basic service set (BSS). A mesh BSS can also be set up, where the STAs participate in forwarding the data towards its destination. Thirdly, multiple BSSs can be interconnected via a wired backbone to establish an extended service set (ESS), in which an STA can seamlessly continue to transmit data after moving from the range of one AP to another following a roaming process.

The standard defines various options for the physical layer that enable ever faster transmission at the cost of higher power consumption. At the time of writing this thesis, speeds of up to 2402 Mbps are possible, with payloads up to 2312 bytes. The frame header is always sent at the lowest speed for reliability and compatibility reasons. The physical layer includes various modulation options, coding techniques, channel bandwidth options and mechanisms for using spatial streams, among other features. Although IEEE 802.11 provides high throughput, the lack of coordination between applications that are part of the same collision domain can easily result in the applications requesting an aggregated throughput that exceeds the transmission capacity of the channel, limiting the ability of IEEE 802.11 to scale well with the number of nodes and data transmissions. The standard defines four MAC protocols: distributed coordination function (DCF), point coordination function mechanism (PCF), enhanced distributed channel access (EDCA) and hybrid coordination function coordinated channel access (HCCA).

DCF is the mandatory and default MAC protocol in IEEE 802.11 (Figure 5.1), featuring CSMA/CA. CSMA/CA is used due to the general limitation of wireless antennas to transmit and receive simultaneously, which prevents them from performing collision detection as with CSMA/CD. Unlike Ethernet, and given the lower reliability of the medium, DCF can use acknowledgment frames (ACKs) to indicate whether a transmission was successful and to retransmit if such an ACK is not received. However, ACKs apply to the exchange of unicast data and exclude broadcast and multicast transmissions. The retransmissions are carried out up to a certain limit. If the limit is exceeded, the data exchange is considered unsuccessful. DCF also counts with the optional RTS/CTS mechanism to combat the problem of hidden nodes. Due to the overhead of the RTS/CTS mechanism, it is sometimes disabled in situations where the hidden node problem is not expected. In summary, data transmission in DCF follows a sequence in which the node that wants to send the data senses the medium for an IFS duration named DCF IFS (DIFS). If no other transmissions are detected, RTS and CTS are exchanged, if enabled. The data to be transmitted is then sent, followed by an ACK. Each of the transmissions in this sequence is separated by an idle time called short interframe space (SIFS). Since the SIFS is shorter than the DIFS, nodes willing to initiate a data exchange cannot interrupt a transmission sequence in progress. Moreover, since other devices know the steps of the sequence, they can calculate when the ongoing transmission will be completed as soon as they receive the information about such a transmission from the data frame header. In this way, they could defer the channel sensing to save energy, a mechanism named network allocation vector (NAV). If sensing the medium before attempting to send data results in the detection of some ongoing transmission, the node backs off, as described in the CSMA/CA protocol. DCF is a mechanism used to enable data transmissions in situations without congestion, with nodes coordinating without requiring a central node. As it is based on CSMA/CA, it is unfortunately not suitable for meeting the requirements of dependability and real time.

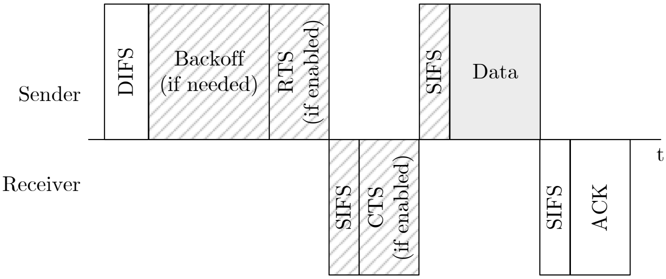


Figure 5.1: IEEE 802.11 DCF data exchange timeline

PCF can provide bounded medium access time based on a polling mechanism. The protocol only applies to STAs that are connected via an AP that acts as a coordinator. In PCF, medium access is divided into periodically repeated phases consisting of two parts: A contention-free period (CFP), that uses polling, and a contention period (CP), which uses DCF. In the CFP, the coordinating node sends polling frames, the contention-free poll frames (CF-Poll), to the other STAs to give them a chance to transmit. The polled node must transmit a null frame, i.e., a frame that contains no data, if it has no data frames to send. The order in which the STAs are polled can be adapted to the user's needs. The time between frames in CFP is given by the PCF interframe space (PIFS), which is shorter than the DIFS, prioritizing frames using PCF over DCF. Although the mechanism provides a bounded channel access time, it has shown some limitations related to the loss of the so-called beacon frame, which is used to trigger the contention-free period [40]. Furthermore, if a STA gains access to the channel, it may occupy it for a non-deterministic time interval. As it is common with polling protocols, this mechanism also causes significant overhead. In addition, PCF suffers from problems with hidden nodes. As a consequence, a standard amendment trying to bring some improvements, IEEE 802.11e, was introduced.

The IEEE 802.11e standard amendment focuses on improvements for delay-critical data exchanges by defining two new coordination functions: EDCA and HCCA.

EDCA is based on a selection of different channel access parameters from CSMA/CA to enable the definition of four traffic classes, named access categories (ACs) in the amendment (Figure 5.2). The proposed categories are, from higher to lower priority: voice (AC_VO), video (AC_VI), best effort (AC_BE) and background (AC_BK). The ACs effectively mean different prioritization when accessing the channel by selecting different values for the DIFS, named arbitration interframe space (AIFS) in the amendment, different boundaries for the CW and the limit in the transmission opportunities (TXOP). The TXOP limit indicates how long several transmissions from a node may continue without sensing the channel once access to the medium has been granted to the node. EDCA also uses eight priorities, coming from IEEE 802.1Q, to classify the data frames, with each of the priorities assigned to one of the four different ACs. The ACs also translate to different queues in the STAs. In this way, frames from different ACs do not interfere with each other in the queues. Due to the customized values for AIFS and CW, the time to access the channel for an AC with a higher priority is reduced on average. However, the default CW values for the access categories partially overlap, which can result in frames from lower priority ACs gaining access before those with higher priority, even if they start the channel access arbitration at the same time. EDCA improves transmission latency, but access to the medium is not deterministic, as there may still be collisions of traffic with the same or different priority [42].

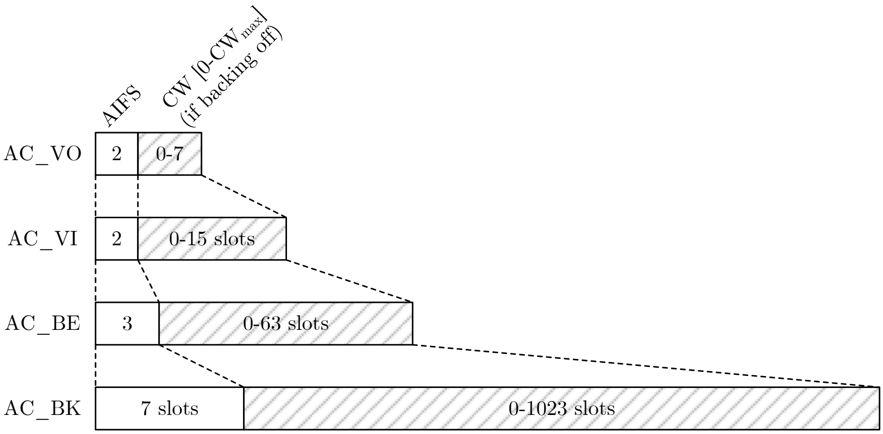


Figure 5.2: EDCA access categories

The HCCA mechanism is similar to the PCF mechanism, but enables the initiation of contention-free phases named controlled-access phase (CAP) within the contention period. With HCCA, the time between two beacon frames is divided into CP and CFP phases, as with PCF, that are repeated cyclically in the so-called superframe structure (Figure 5.3). The CAP phases can also be initiated during the CP, but only up to a certain limit, so that it is guaranteed that there is still some time for contending transmissions, which are ruled by EDCA. The scheduling of TXOPs in HCCA is not addressed in the standard, leaving it open to the implementer. HCCA can provide deterministic data transmissions. However, its polling mechanism leads to a significant communication overhead, especially when transmitting frames with small payload [41]. Currently, COTS devices do not implement HCCA and it is unclear if and when the industry will adopt HCCA in the future.

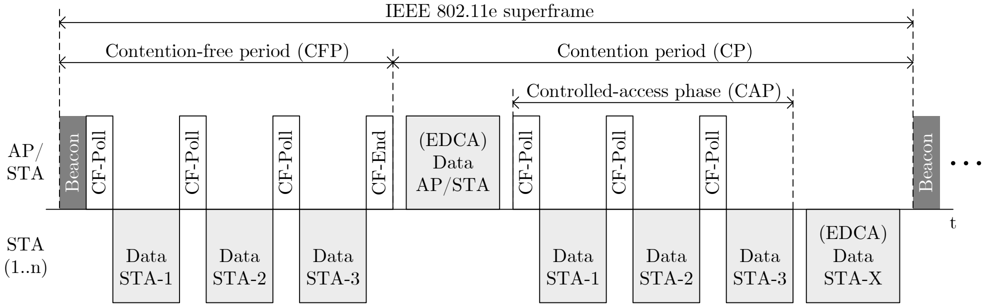


Figure 5.3: Example of an IEEE 802.11e MAC superframe timeline

The CP and CFP mechanisms and the polling scheme of PCF and HCCA allow different treatment of traffic with different criticality and enable the protection of data exchanges from different applications. In IEEE 802.11e, the access is subject to admission control after STAs associate with an AP. The STA specifies its data transmission requirements in the so-called TSPEC frame and the AP allocates them if sufficient resources are available. The AP then creates the polling schedule. Such negotiation supports STAs leaving and joining the network dynamically, which favors the adaptability of the technology.

The IEEE 802.11 standard includes an additional mechanism called block acknowledgment, where multiple transmissions are acknowledged at once instead of sending a single ACK for each data transmission. The MAC layer also takes care of other aspects such as addressing or data integrity mechanisms. In addition, IEEE 802.11 uses the timing synchronization function (TSF) to keep the STAs synchronized to the AP based on the exchange of the beacon frame, but the mechanism has documented issues [43] related to a higher probability of losing synchronization as the number of nodes increases. Other standard amendments, like IEEE 802.11p, include mechanisms aimed at applications in vehicular data transmission networks.

5.2 Operational technology

The digitalization of functions, e.g., after the introduction of ECUs in automotive or programmable logic controllers (PLCs) in industrial environments, prompted the need for deploying data communication networks. At the time of this early adoption, there was not a common reference dependable and real-time solution that could be used for the various scenarios in industrial automation, aerospace, automotive or robotics. Thus, the fulfillment of the demanding requirements led to the development of various proprietary technologies, which were later labeled under the term OT. The numerous proprietary technologies led to barely compatible deployments, which were often based on expensive equipment that could not keep up with the throughput improvements of IT technologies such as Ethernet. A selection of wired- and wireless-based OTs is provided next in order to get an overview of what is required to integrate IT into OT and wireless into wired deployments.

5.2.1 Overview of wired protocols

Wired deployments are the default option for supporting dependable and real-time requirements in the OT field, unless other aspects such as mobility make them opt for wireless options.

Controller Area Network (CAN)

CAN (ISO 11898-1 [19]) is a wired technology initially designed for the interconnection of electronic components in the automotive domain, which remains its most popular application area. It covers the physical and data-link layers. Later, several upper layer protocols were developed to facilitate the use of CAN in various applications. The networks are deployed following a bus topology so that the transmission medium is shared by all end systems involved. The shared medium as well as the limited throughput of 1 Mbps and the small payload of 8 bytes make CAN a technology that is not suitable for large and fast data exchanges, limiting its scalability. A revision of the protocol, CAN Flexible Data Rate (CAN-FD), focused on improving throughput but remained quite limited, increasing it to 5 Mbps with payloads up to 64 bytes.

The MAC protocol is based on a distributed approach in which a node attempts to transmit as soon as it has data. The transmission is contention-free but is only granted after an arbitration phase based on a priority assigned to the nodes in advance. The use of priorities allows the network infrastructure to be shared for the exchange of data of different criticality, with the limitation that such differentiation is only made on a node-by-node basis and not for a specific data transmission. Thus, it is possible to differentiate and prioritize some data transmissions from others, but only if they originate from different nodes. For the exchange of bounded amounts of data, it is possible to analyze the behavior of the arbitration protocol and calculate an upper bound for the channel access time, enabling the fulfillment of deadline requirements [44]. Among the available upper-layer protocols, Time-Triggered CAN (TT-CAN) allows to deploy a static time-triggered schedule for deterministic access to the medium in combination with the built-in event-driven access. In CAN networks, the configuration of priorities is static, so the real-time behavior cannot change at runtime, which limits its adaptability. Reliability is supported by providing timing guarantees, a reliable medium and retransmission mechanisms based on time redundancy after an unsuccessful data ACK.

CAN does not offer timestamping and synchronization directly but can be achieved with other protocols that run over CAN. Although CAN is still used in automotive deployments, it has been partially replaced or complemented by FlexRay and MOST due to the increase in drive-by-wire and multimedia applications, respectively. The latest trends show an increasing interest in the introduction of TSN in the automotive industry.

FlexRay

The introduction of FlexRay (ISO 17458-1 to 17458-5 [45]) as a wired protocol for automotive was primarily targeted to overcome the limitations of the CAN bus. Due to these limitations, CAN was not able to support drive-by-wire systems where the mechanical components do not completely assume the control of the vehicle, but computer systems and possibly networks take over part of the responsibility. FlexRay covers the physical, data-link and application layers. The protocol transfers data at up to 10 Mbps with a payload of 254 bytes, which is significantly larger than CAN. The data rate and payload provide better but still limited scalability to support a larger number of nodes and transmissions. FlexRay deployments adopt both bus and star topologies.

Each of the segments that the topology defines is shared by several nodes that access them via the MAC protocol, which divides the available time into slots. The time slots are grouped into static and dynamic phases that repeat cyclically. The static phase defines a contention-free MAC protocol with bounded-access time that allows time-triggered data transfers with unambiguously assigned time slots. The dynamic phase supports event-driven data exchanges with a contention-based protocol where access is controlled on the basis of priority. The dynamic phase is divided into minislots to which an index is assigned. The data sent during the dynamic phase is also assigned an index, which serves as a means of prioritization. When the dynamic section is entered, the devices set their counters to point to the first minislot. If there is data that matches the index of the current minislot, that data is sent. The data may require several minislots until the exchange is complete. The index increases as soon as the next minislot begins if there is no data matching the current index or a transfer has been completed. This mechanism prioritizes data with lower indexes, potentially starving lower priority transmissions. To ensure that the FlexRay nodes have the same view of the timing of the slots, they are synchronized. The mechanisms of pre-assigned time slots in the static phase and prioritization of data in the dynamic phase allow supporting data with different criticality. These mechanisms also help in partitioning network resources between data exchanges coming from different nodes or from the same node. The allocation of time-triggered slots or the prioritization of data in the dynamic phase requires scheduling algorithms specifically designed to meet the timing requirements [46]. Reconfiguration of the static and dynamic phases during FlexRay's runtime is not supported, which limits its adaptability. However, some implementations can reconfigure the dynamic phase without having to reset the network components.

The protocol supports redundant medium for fault tolerance.

Avionics Full-Duplex Switched Ethernet (AFDX)

AFDX (ARINC 664 part 7 [23]) is a wired data communications standard from the aircraft industry based on switched Ethernet. Its main application is fly-by-wire systems, which, similar to drive-by-wire systems in the automotive industry, aim to control the aircraft using computers instead of relying solely on traditional hydromechanical control components. The AFDX standard covers the four bottom layers of the OSI model. For the physical and data-link layers, the protocol is based on Ethernet, but with the addition of the virtual link (VL) mechanism in the data-link layer. VLs are used to establish bandwidth-guaranteed logical paths between a sender and a set of receivers via the physical star topologies. The network layer is based on a restricted version of IP, while the transport layer is based on the user datagram protocol (UDP). The adoption of the UDP/IP stack facilitates the integration of other upper layer protocols that are often used together with IP. With a physical layer based on Ethernet, speeds of up to 1000 Mbps can favor scalability. The payload at the physical layer can reach up to 1500 bytes.

The MAC protocol is based on CSMA/CD as defined by Ethernet, but the use of VLs over switched Ethernet enables guaranteed latencies with relatively low jitter. Bandwidth reservation for the VLs is performed via the leaky-budget algorithm. Such bandwidth reservation takes place offline and cannot be reconfigured dynamically, which has a negative impact on adaptability. VLs also provide means to partition the network resources between data transmissions with different criticality, coming from different nodes or sending applications, so that the impact of data exchanges on each other is limited. To provide real-time guarantees, response-time analysis can be used [47].

AFDX comes with built-in redundancy by replicating the entire network, including links and network devices. Furthermore, the asynchronous data handling removes the need for clock synchronization.

Highway Addressable Remote Transducer (HART)

HART is a wired communication technology for industrial automation that was introduced in the 1980s. It is an excellent example of a technology that is still used in many deployments despite its introduction decades ago. Its current presence is due in part to its compatibility with existing industrial infrastructure, the significant cost of replacing it with newer technologies, and the various improvements HART has undergone, including the physical-layer-agnostic version, HART-IP, and the wireless version, WirelessHART. Only the basic version is covered here, but details of WirelessHART can be found later as part of the wireless OT technologies. HART comprises the physical, data-link and application layers. The signals can be transmitted in digital or analog format at a very slow rate of 1.2 kbps, which does not favor the scalability of the deployments. The supported payload reaches up to 254 bytes. Another limitation for scalability is that a maximum of 15 nodes are supported in a HART network following daisy chain or star topologies. The interactions between the nodes are constrained to a master-slave[1] relationship, with a maximum of two masters per deployment. The masters can issue commands to the slaves, which answer them according to a request-response protocol. Alternatively, the slaves can send updates to the master at regular intervals without polling in the so-called burst mode.

The MAC can provide bounded latency to access the medium, and transmissions are contention-free as the master and slaves take turns accessing the medium in burst mode. The rules for when each device can access the network depend on the number

[1] The use of master-slave terminology is in decline as new technologies adopt other terms. The new terms are also based on analogies to human interactions, but are no longer reminiscent of slavery, e.g., publisher and subscriber.

of masters and the number of slaves in burst mode. The commands in HART are predefined interactions that respond to specific actions, e.g., requesting a sensor value from a slave. HART does not treat data originating from different applications or of distinct criticality differently. The MAC protocol requires response-time analysis to prove that it can provide bounded latencies when accessing the medium, which depend on the selected transmission mode and the number of connected nodes. The configuration of the MAC, in turn, is not complex, apart from the selection of the transmission mode, which depends on how frequently updates are to be provided. However, the configuration is done offline, which limits its adaptability.

HART does not provide support for fault-tolerant mechanisms and data timestamping.

PROFIBUS

The industrial wired communication technology PROFIBUS (International Electrotechnical Commission [IEC] 61158 [18]) covers the physical, the data-link and the application layers. PROFIBUS comes in two physical layer variants, yet the maximum transmission rate of 12 Mbps and the data transmissions of up to 246 bytes do not promote scalability. The nodes in a PROFIBUS network are deployed in daisy-chain, ring or star topologies. The nodes follow master and slave roles, with the slaves typically being sensor and actuator devices.

As the network can have several master nodes controlling the slaves, a logical ring is set up between them to establish a token-passing protocol. The MAC protocol ensures a contention-free bounded-time medium access. Holding the token gives the master the right to initiate various data exchange interactions: Send Data with No acknowledge (SDN), Send Data with Acknowledge (SDA), Request Data with Reply (RDR), and Send and Request Data (SRD). Data is labeled as either high or low priority in PROFIBUS and stored in separate queues before transmission, with low priority data only being transmitted if no high-priority data is available. The prioritization mechanism is used to support flows of different criticality. It is possible to prove a bounded access time to the medium using response-time analysis [48], making PROFIBUS suitable for real-time deployments. In addition, the token-holding mechanism helps to ensure that different data exchanges share the network but limit the mutual impact. PROFIBUS is configured offline using special software that is used to describe the different nodes in a pre-defined format. Offline configuration rules out dynamic reconfiguration and impacts adaptability negatively.

The protocol supports clock synchronization and timestamping. In addition, fault-tolerant mechanisms can be applied to have redundant master or slave nodes, or an additional bus for medium redundancy.

PROFINET

PROFINET (IEC 61158 [18] and IEC 61784 [22]) is an Ethernet-based industrial wired communication protocol designed as a successor to PROFIBUS. Its specification deals with aspects of the physical, data-link, network, transport and application layers. Based on Ethernet, speeds of up to 100 Mbps with payloads of up to 1500 bytes at the physical-layer level are possible, which favors scalability. The network between the nodes is deployed following daisy-chain, ring, star and tree topologies, with the latter using switches to connect the nodes to each other. The end devices in PROFINET take on three roles: IO devices, IO controllers and IO supervisors. IO controllers are responsible for the configuration of IO devices, with the latter used as regular sensor or actuator nodes. However, these roles are not relevant when establishing communication, e.g., an IO device can provide some data to an IO controller and vice versa. In the case of IO supervisors, these are nodes that are used for monitoring and diagnostics.

With PROFINET, the medium is accessed via four approaches: isochronous real-time (IRT), real-time (RT), TCP/UDP over IP and, more recently, PROFINET over TSN. IRT can achieve very low latencies without contention by scheduling data transfers over time slots and maintaining a common notion of time using clock synchronization based on IEEE 1588 PTP [49], which is described later in the TSN section. RT, on the other hand, does not offer as low jitter as IRT, but can still provide bounded latencies that are free of contention. The coexistence of IRT, RT and TCP/UDP over IP is based on the division of the available time into two phases, which are repeated cyclically. One of the phases is dedicated solely to IRT, while the other is shared by RT and TCP/UPD over IP. RT can keep its time guarantees as it is firstly identified with the IEEE 802.1Q VLAN tag and secondly prioritized. In this way, RT is not impacted by TCP/UPD over IP data exchanges, which could follow an unbounded traffic pattern. In the case of TSN, PROFINET does not rely on IRT and RT to fulfill the timing requirements, but on a set of selected TSN mechanisms to ensure low-latency jitter free of contention. Further details on TSN can be found later in the TSN section. PROFINET is complemented by an application layer that applies to all access methods. A PROFINET network with only RT devices does not require any changes to the Ethernet stack, while the option of an IRT phase requires special hardware. Further, due to the relaxed timing requirements, RT can operate over wireless networks such as IEEE 802.11 and IEEE 802.15.1 (Bluetooth). Thanks to the definition of the different transmission modes, it is possible to support data with different criticality. The response-time analysis can be performed to get figures on the actual real-time behavior [50]. The configuration of PROFINET is done offline with engineering tools, so dynamic reconfiguration is excluded, which severely limits adaptability.

Redundancy can be applied at different levels, including IO controllers, IO devices, medium and alternatively the entire network, including medium and intermediate devices.

POWERLINK

The industrial Ethernet-based wired communication technology POWERLINK offers a customized solution for part of the data-link and application layers. It leaves the physical and part of the data-link layers to Ethernet and includes TCP and UDP over IP for the transport and network layers, respectively. POWERLINK offers speeds of up to 1 Gbps and transmissions of up to 1500 bytes of physical-layer payload, which favors scalability. Network equipment based on Ethernet does not need to be modified as the POWERLINK is fully implemented in software. The protocol also supports a variety of topologies derived from Ethernet, i.e., daisy chain, ring, star and tree.

Like other previously examined protocols, POWERLINK divides the available transmission time into isochronous and asynchronous phases, which are executed cyclically. In the isochronous phase, a managed node (MN) polls the set of controlled nodes (CNs) in a round-robin sequence, providing a bounded and contention-free medium access method. In the asynchronous phase, access to the medium is granted using an unmodified Ethernet MAC. The asynchronous phase allows hot plugging of CNs which can communicate with the MN until their access is granted in the isochronous phase. The asynchronous phase can also be used by external devices that do not run the POWERLINK stack but get access to the POWERLINK network via a gateway. Regardless of the data handling mechanism, the nodes exchange data following three types of relationships: Master/slave, client/server and producer/consumer, which differ in who has the initiator role and how the exchange interaction is performed. The two available data handling mechanisms support data exchanges of different criticality. In addition, the asynchronous phase enables the partitioning of network resources so that data exchanges can be performed without affecting the isochronous phase. Theoretical analysis is used to ensure the real-time behavior of data exchanges [51].

POWERLINK supports clock synchronization via IEEE 1588. Redundancy is enabled by having several MNs, and also on the medium level by using a ring topology.

Ethernet for Control Automation Technology (EtherCAT)

EtherCAT (IEC 61158 [18], IEC 61784-2 [52] and IEC 61800-7 [53]) is an Ethernet-based industrial wired network technology that does not require any modifications to the Ethernet hardware for operation. However, better timing performance is achieved with special hardware support. EtherCAT offers a tailored solution for part of the data-link and application layers. The physical and part of the data-link layer are based on Ethernet, and TCP or UDP over IP cover the network and transport layers. EtherCAT can transfer up to 1500 bytes of physical-layer payload at a rate of up to 100 Mbps, allowing the technology to scale with the number of data transmissions. Supported topologies in EtherCAT include daisy chain, ring, star and tree.

EtherCAT features token-based medium access, with the particularity that the token is an Ethernet frame whose sending is triggered by a master node and then forwarded from one slave to another. The payload of the frame is divided into several fields, each of which corresponds to a node that updates it on receipt. Once the frame has visited all the slave nodes, it begins its return journey in the opposite direction, towards the master. Since the data from sensors and actuators in industrial processes is usually relatively small, EtherCAT delivers good performance even when having many devices by avoiding the overhead of using different frames for small payloads but merging all data into a single one instead. In turn, applications with long transmissions will probably not perform well. As for the real-time properties, the time needed to receive an update of the status of the slaves is bounded. The bounded time is possible because the token is sent periodically by the master and the path to retrieve the slave status is known. In addition, the transmissions are contention-free. Partition of network resources so that different data transfers do not interfere with each other is not supported, nor is the support for data with different criticality, as all transfers are treated equally. The real-time analysis of the protocol can detail the performance of EtherCAT concerning timing requirements [50]. The configuration of an EtherCAT network is based on offline tooling. Such tooling does not support dynamic reconfiguration, which has a negative impact on adaptability.

The technology includes clock synchronization and timestamps. Redundancy is available for the medium subject to each component counting with a second network interface.

High-availability Seamless Redundancy (HSR)

Compared to the OT protocols considered up to now, HSR (IEC 62439-3 [54]) does not add real-time capabilities, but its focus is on enabling fault tolerance on Ethernet. The protocol is primarily applied to industrial use cases. HSR uses Ethernet for the physical and data-link layers, but modifies the data-link layer to add fault-tolerance support. With HSR, the nodes count with special hardware and attach themselves to a ring network via two ports. As the links are full-duplex, there are two paths to reach a destination. Both paths are used in parallel by duplicating the frames and making them circulate in parallel in opposite directions. A failure of one path therefore requires no additional time for recovery. After the first copy of a frame arrives at the destination, duplicated frames are discarded on receipt based on the identification information provided as part of the HSR frame. Unfortunately, adding nodes in a ring topology increases latencies, which limits scalability. In addition, such an addition requires the reconfiguration of the ring. The reconfiguration can lead to an interruption of the service, which limits its adaptability.

Parallel Redundancy Protocol (PRP)

PRP (IEC 62439-3 [54]) is used like HSR to enable fault-tolerant Ethernet in industrial environments. The protocol relies on the physical and data layers of Ethernet, but modifying the latter. Fault tolerance is achieved by connecting each node to two independent networks that can take any topology, which favors scalability. Data is sent in parallel over both networks, which requires special hardware support. As the data is transmitted via redundant paths, a failure in one path can be immediately covered by the other. Duplicate frames are discarded at the destination based on the identification information encoded as part of the PRP frame. When adding or removing a node, the reconfiguration of the redundant networks can lead to service interruptions, which limits the adaptability of the system.

Media Redundancy Protocol (MRP)

MRP (IEC 62439-2 [55]) is another alternative to provide Ethernet with fault tolerance, also targeting industrial environments. The protocol applies to the data-link layer and uses the physical and data-link layers of Ethernet. MRP uses specialized hardware to connect each node to a ring network via two ports. A node is selected as the ring manager and blocks one of its ports so that the ring remains open at this specific location. Data is sent via one of the ports from a node selected based on the destination. Unlike HSR and PRP, MRP provides fault tolerance through a cold standby mechanism. If a problem occurs in the ring, the ring turns to be open at two points, dividing the network into two segments. The ring manager then activates the previously blocked port to re-establish the connection between the two segments. As in the case of HSR, the ring topology does not scale well, as adding a new node directly leads to increased transmission latency. In addition, the required reconfiguration when adding a new node may result in a service interruption, which has a negative impact on adaptability.

Time-Triggered Ethernet (TTE)

TTE (SAE AS6802 [24]) extends Ethernet with both guaranteed and bounded latency communication for applications with different criticality. The protocol was primarily used in the aerospace industry, but is also present in the automotive sector. TTE operates on the data-link layer and uses the physical and data-link layer of Ethernet. TTE devices achieve speeds of up to 1000 Mbps with payloads up to 1500 bytes at the physical-layer level. TTE networks are set up with special switches, with end systems opting for hardware or software support depending on the tightness of timing requirements. The interconnection of end systems and switches is not restricted, and ring, star or tree topologies are possible. The high throughput, together with the topology freedom favor the scalability of the system.

TTE support for mixed-criticality and different timing guarantees is founded on three traffic classes: time-triggered (TT), rate-constrained (RC) and best effort (BE). The TT traffic class fulfills the strictest dependability and real-time requirements, enabling contention-free periodic message delivery with very low jitter. The RC traffic class is inherited from AFDX and provides bandwidth guarantees by transmitting data in a contention-free manner as long as the data is not generated more frequently than the MIT specifies. Finally, the best-effort traffic class offers no delivery guarantees, as contention might occur in the switches' memory. BE is used to channel data as in a regular Ethernet network, which enables access from end systems without TTE support. Traffic prioritization is performed on the basis of dataflows, so that critical, delay-sensitive traffic is fully deterministic, while low-priority traffic can suffer from starvation. TT and RC traffic classes are based on the concept of VL as a unidirectional dataflow from a sender to a set of receivers that follows a periodic or sporadic exchange pattern. Differentiation via traffic classes and bandwidth reservation on a VL basis work as a resource partitioning mechanism, which enables data transmissions of different criticality without compromising highly-critical traffic. The protocol requires the generation of a schedule configuration based on the communication requirements of the applications. The schedule, calculated offline, is passed to the switches and end systems that form a TTE network to make them aware of the times at which they must take TT messages from the buffers and forward them to the output ports. The same principle applies to the times at which TT messages are expected to be received. For RC messages, the scheduler ensures an average bandwidth, which means that TT messages leave enough unused time to guarantee the RC bandwidth. In the worst case, RC traffic is periodic, so that the scheduler can regularly leave free slots after allocating TT slots. BE messages are placed during runtime in the free slots left by TT and RC traffic. Since the configuration is created by a network engineering tool that provides it to the network nodes before operation starts, dynamic reconfiguration is ruled out, causing adaptability to be greatly impaired. The schedule has a specific duration, after which it is repeated cyclically. The work in [11] proposes a mechanism to solve the FOL constraints for scheduling using SMT solvers. In contrast to the usual FIFO memories at the switches, TTE includes a buffer structure where the first frame from each VL can be selected even if other frames from other VLs arrived earlier at the switch. Buffering is particularly advantageous to improve the network's schedulability and better meet its timing requirements. In addition, the three traffic classes are divided into different buffers. To enable the nodes to adhere to the schedule and event timestamping, TTE has a built-in clock synchronization protocol that enables a precise timing for the events. TTE uses the same frame format as Ethernet, but encodes the identification of the VLs in the destination MAC address.

TTE supports redundancy through replication of the network, the medium or at the device level.

Flexible Time-Triggered Ethernet (FTT-E)

FTT-E [56] is a network technology applicable to the data-link layer that relies on the physical layer of Ethernet. The technology has not yet made it to commercial products, but offers interesting features. FTT-E attempts to overcome the rigidity of offline scheduling that is common with time-triggered networks. To do this, it offers an online admission control mechanism that makes it possible to dynamically add or remove traffic flows and adapt the schedule to them. The so-called elementary cycle is divided into synchronous and asynchronous windows. In the synchronous window, an initial message is used to transmit the schedule for the messages in the remaining part of the window, providing a contention-free and bounded-access MAC scheme. In the asynchronous window, a central coordinator polls the nodes for the presence of event-driven messages. The polling sequence is defined by the scheduling policy decided by the user and the analysis of the real-time properties is left open. Since FTT-E is based on Ethernet, it is able to support the rates and payload of Ethernet. The devices are arranged in star topologies. The partition of transmissions into synchronous and asynchronous phases enables transmissions with different time and criticality requirements.

FTT-E does not require clock synchronization to function. Redundancy is supported by having multiple coordinator nodes.

Time-Sensitive Networking (TSN)

TSN [25][26][27] is a set of standards that describe mechanisms to enable dependability and real-time guarantees over IEEE 802 technologies. TSN is applied at the data-link layer. Its main target is to work with Ethernet, but TSN is also applicable with potential limitations to other IEEE 802-related technologies, including IEEE 802.11. When applied over Ethernet, speeds of up to 10 000 Mbps and data transfers of up to 1500 bytes of physical-layer payload are achievable over switch-based topologies such as ring, star and tree, which has a positive impact on scalability. Unlike the previously reviewed protocols, which might be offered in multiple ready-to-use variants and technology versions with different features, TSN does not specify a prescribed set of its mechanisms to be provided by default. Instead, users are expected to select the TSN mechanisms that apply to their use case based on the requirements. When making such decisions, not only real-time and dependability aspects need to be considered, if any, but also others such as the availability of the mechanisms on the switches. In addition, some mechanisms must be implemented in hardware to achieve the level of performance required by the respective TSN mechanism specification, while others can remain primarily in software. The mechanisms can theoretically be used in any combination. In practice, however, the different scenarios result in subsets of mechanisms that can be better used together and that are to some extent interdependent. Such an open approach to the selection of mechanisms has triggered several initiatives coming from different domains and trying to define use cases and possible solutions for the combination of TSN mechanisms. The IEEE standardization organization hosts some of these efforts with the definition of the so-called TSN profiles for industrial, automotive or aerospace, just to name a few.

TSN uses the concept of streams, similar to VLs, to define unidirectional logical data connections over the physical topology between a sender and a set of receivers. The data is transported in Ethernet frames that contain the IEEE 802.1Q VLAN tag. The VLAN tag features the priority code point (PCP), which can be used to assign eight different priorities to frames.

The handling of frames in the switches (Figure 5.4) is divided into three stages: ingress, switching and egress. In the ingress stage, the frame is received via the inbound port. The data streams are then identified and can be filtered, i.e., discarded, on the basis of predefined rules. In the ingress stage, fields of the frame can also be temporarily modified within the switch or permanently so that they remain outside the switch domain. Such alteration becomes handy to add missing handling information, e.g. if the end system cannot add the VLAN tag itself. It also allows to assign a frame to different queues on each switch. This added flexibility increases the probability of finding a schedulable system. Then, if the frame is not discarded, it goes to the switching engine where it is forwarded to the appropriate outgoing port based on the frame's destination MAC. At the outbound port, the frame goes through the egress stage where it is enqueued into eight different FIFO queues based on the frame's PCP value. At this point, the frames can be selected for transmission from the queue based on the gate status. The flow of frames is then subject to being shaped with different mechanisms. It is relevant to note that a common misconception is to associate PCP with criticality. PCP should rather be seen as a way of assigning data streams to different memory locations, i.e., queues, on the switch. PCP only serves as a prioritization mechanism if several gates allow the selection of frames for transmission in the same time interval. In this case, the stream with the highest priority is selected first. Overall, the PCP and shaping mechanisms enable the handling of data transmissions with different criticality and the partition of network resources so that different data transfers do not influence each other, subject to a configuration that is correct. TSN mechanisms allow to use different filtering and shaping for the streams based on their identification. The availability of TSN mechanisms creates many possibilities for their combination and makes it very complex to create a configuration that meets the requirements of data exchanges. Specifically, the configuration of data scheduling is NP-complete in terms of computational complexity [57]. Although dynamic reconfiguration is one of the mechanisms supported by TSN, generating configurations at runtime can be challenging and affects adaptability.

TSN mechanisms are configured via so-called managed objects, which can be read

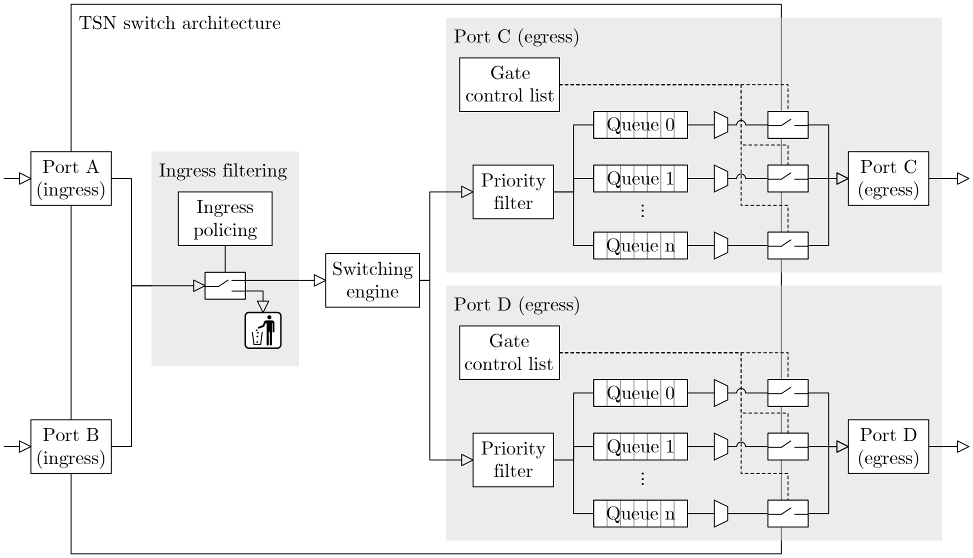


Figure 5.4: Architecture of a TSN switch

and modified using configuration protocols. However, the configuration protocols and generation are not defined as part of the TSN standards.

A selection of TSN standards and their mechanisms are presented and described below.

"Timing and Synchronization" (IEEE 802.1AS [25]). Provides network devices with a common notion of time that guarantees a maximum time difference of 1 μs for devices that are seven links apart. The standard defines the generalized precision time protocol (gPTP) as a subset of PTP (IEEE 1588 [49]) and includes support for clock redundancy and multiple time domains.

The network devices participating in the synchronization count with a PTP clock. Initially, these clocks run independently and drift from each other. When the synchronization protocol is running, the clocks expose themselves via the ports of the network devices and assume the role of either the master or the slave. A master port provides the device's PTP clock to the slave at the other end of the link. The aim of the synchronization protocol is to adjust the slave clocks so that the difference to the master is reduced to be lower than the requested defined precision. A device can therefore have several ports, of which at most one receives the time and the others supply it. In a deployment with multiple master ports, only one clock, i.e., the grandmaster, is selected as the source for all clocks. The grandmaster can only be located in one end system and is selected manually or via the best master clock algorithm (BMCA).

The synchronization protocol exchanges the value of the master clock with the slaves so that they can correct their clocks. However, the value cannot be used directly, but must be updated taking into account the time it took to travel from the master to the slave. The standard includes two options for the exchange of synchronization information that also include the measurement of the delay between nodes: peer-to-peer (P2P) and end-to-end (E2E). P2P works between two nodes that are connected via a link, while E2E can involve multiple intermediate nodes and is therefore affected by aspects such as the message queueing. Therefore, P2P is often preferred when all network switches are PTP-capable. The frequency at which the information exchange is triggered is determined by the synchronization interval parameter, with more frequent updates being better to reduce the synchronization offset but increasing the protocol overhead. The protocol is supported by timestamping of messages, which is often done directly in hardware to achieve better precision. Clock correction on the slaves is usually done by increasing or decreasing the clock rate so that the trend of the clocks changes from being initially divergent to converging in small steps. Adjusting the clock rate is often better than simply setting the clock to a new value, as the latter could involve a drastic change that could have undesirable consequences for the handling of timing events on the devices.

In addition, the PTP clocks in the switches can be classified into two categories depending on how they handle the incoming synchronization data: transparent or boundary. With the transparent clock, the received synchronization message is updated with the time spent on the switch before forwarding, the so-called residence time. In contrast, the boundary clock is corrected with the values received at the slave port of the switch. As soon as the value has been corrected, it is passed on to the slave nodes connected to the switch in the role of the master.

Regarding the differences between PTP and gPTP, the gPTP profile restricts the PTP options, and only allows systems with PTP clocks and P2P delay measurement.

"Enhancements for Scheduled Traffic" (IEEE 802.1Qbv [26]). Enables to control the time at which each of the eight queues located on the egress ports selects its frames for transmission. When the transmission selection mechanism is properly configured, data exchanges of the so-called scheduled traffic (ST) occur with guaranteed latency, low jitter and free of contention. The selection mechanism is controlled by a transmission gate (TG) that can take two states, open and closed, which means that frames from the corresponding queue are selected for transmission or not, respectively. The schedule describing the state of the TGs at a given time is encoded as part of the gate control list (GCL). When the end of the GCL is reached, the list is repeated again. If several frames are selected for transmission simultaneously by the GCL, i.e., several TGs are open at the same time, the priority from the PCP is used to decide which frame is sent first. In order to have network-wide coordinated GCL schedules, clock synchronization as provided by IEEE 802.1AS is required. Generating a suitable configuration for the GCL enables ST to meet its timing requirements. However, obtaining such a scheduling configuration is challenging as several clashing factors need to be considered, including stream timing requirements, the limited number of egress queues and the FIFO structure of the queues. Finding a solution becomes even more challenging when multi-hop switched topologies must be considered, as opposed to the simplicity of bus or star topologies. How to solve the scheduling problem is not covered by the TSN standards, but fortunately there are already solutions [57].

"Forwarding and Queuing Enhancements for Time-Sensitive Streams" (IEEE 802.1Qav [26]). Describes the credit-based shaper (CBS) as a mechanism for reserving bandwidth, which enables contention-free data exchanges with bounded latency. If available, CBS is applied on a queue basis to smooth the exchange of bursts of data frames by spacing apart the sending of frames from each other. Following the concept of sporadic servers, CBS is built around the value of the credit. If the credit is positive, frames can be selected for transmission. The credit decreases when frames are sent. If no frames are sent, the credit increases up to the credit limit.

"Asynchronous Traffic Shaping" (IEEE 802.1Qcr [26]). The queued frames at the egress ports can be selected for transmission according to different asynchronous shaping, e.g., the urgency-based scheduler or the Paternoster scheduler. These shaping mechanisms are designed to smooth traffic patterns without the need for network-wide scheduling or synchronization. Analysis tools are required to evaluate their timing performance. Nevertheless, the asynchronous shaping does not aim for as low latency jitter or high bandwidth utilization as ST [58].

"Frame Replication and Elimination for Reliability" (IEEE 802.1CB [27]). The frame replication and elimination feature enables redundancy using space diversity by creating copies of the same data and sending them via redundant paths. The logical paths are established over the physical topology using "Path Control and Reservation" (IEEE 802.1Qca). IEEE 802.1Qca calculates the shortest route between each pair of switches. If several routes have the same cost, a copy of the data frame is sent on each of them. With the correct setting of this mechanism, it is possible to perform simultaneous transmissions over paths with the same cost or to enable backup routes if the primary route, i.e., the one with the lowest cost, is not available. The replicated frames are tagged with the so-called R-TAG, which includes a sequence number that can be used to distinguish the different iterations of frames in a stream. Once the redundant paths for the replicated frames merge, the R-TAG is used to identify duplicate frames and discard redundant copies. The standard amendment also includes stream identification as a mechanism to identify a stream based on the content of the frames belonging to the stream. The identification is based on a unique combination of selected layer 2 parameters such as source and destination MAC, PCP, VLAN ID or EtherType. Upper-layer fields such as an IP address can also be used for identification. Identification can be passive or active. With passive identification, the frame fields are examined without changing them. In contrast, active identification can permanently change fields of the frame, e.g., to add a VLAN tag to a frame. The active identification mechanisms are not to be confused with IEEE 802.1Qci's ability to temporarily change some fields while the frame is in the switch. The stream identification is crucial to enable the filtering and policing features of IEEE 802.1Qci, which are introduced next.

"Per-Stream Filtering and Policing" (IEEE 802.1Qci [26]). The mechanism is intended to filter frames that belong to streams at the ingress stage. Such filtering can result in frames to get discarded if they do not meet predefined criteria, e.g., if they do not comply with a maximum rate for a stream or arrive outside a predefined time window. The aim of the mechanism is to provide protection against end systems and switches that do not adhere to their data exchange specification, against nodes that are outside the engineered part of the network or against faulty nodes. The mechanism ensures that such problems are isolated so that they do not affect the entire network. As part of the filtering and policing mechanism, streams are first identified using the stream identification function of IEEE 802.1CB. The identification can be used to decide in which queue and with which flow meter the stream should be handled. Such behavior overrules the enqueuing based on the PCP value and could serve to apply different enqueuing rules on different switches, which might increase the schedulability of data exchanges. The frame then passes through a gate corresponding to its PCP value. The gate is used to decide whether a frame should be allowed through or discarded based on a GCL as defined in IEEE 802.1Qbv. Finally, a flow meter algorithm, e.g., CBS, decides whether a frame is finally admitted into the switch and completes the ingress stage, or whether it is dropped instead.

"Frame Preemption" (IEEE 802.1Qbu [26]). As part of the frame preemption mechanism, frames are classified as preemptable or express, e.g., depending on how critical their transmission is. The mechanism is used to increase throughput performance and improve latencies by pausing the ongoing transmission of preemptable frames when express frames are ready for transmission. The remainder of the paused preemptable frame is sent as soon as no express frames are being transmitted. Without the frame preemption mechanism, a low-priority frame could be sent only when its transmission fits completely without affecting the high-priority frames. So if there is no preemption, there would likely be gaps without transmissions, which would impact the latency for low priority frames negatively.

"Path Control and Reservation" (IEEE 802.1Qca [26]). The path control and reservation mechanism enables the creation of paths between pairs of nodes. The mechanism also supports redundancy to improve reliability. It relies on the SPB routing protocol (IEEE 802.1aq) to calculate the shortest route between a pair of switches. SPB is based on the information provided by the Intermediate System - Intermediate System (IS-IS) link-state protocol encapsulated in Ethernet frames. IS-IS is used by each switch to determine the topology of the network and to exchange information about the routes cost. Initially, the switches exchange IS-IS Hello messages to discover the neighboring switches and determine changes in the topology. Hello messages only provide a view of the adjacencies, while Sequence Number Protocol Data Units (SNPs) and Link State Protocol Data Units (LSPs) are used to propagate the information. Information is exchanged until all switches have the same view of the network topology. This information is stored on each node in Link State Databases (LSDBs). As soon as convergence is reached, SPB sends the data messages on the route with the lowest costs.

"Stream Reservation Protocol (SRP)" (IEEE 802.1Qat [26]). It is a mechanism for reserving bandwidth to enable streams with bounded latency, e.g., for audio and video streaming applications. The potential stream sources, referred to as talkers in the standard, advertise their streams. The potential recipients, referred to as listeners, subscribe to them, establishing a path between them with guaranteed bandwidth. The reservation process is made dynamically and subject to an admission control mechanism that checks that predefined bandwidth thresholds are not exceeded. The CBS mechanism of IEEE 802.1Qav is responsible for enforcing the reservation.

"Stream Reservation Protocol (SRP) Enhancements and Performance Improvements" (IEEE 802.1Qcc [26]). Extends the SRP mechanism of IEEE 802.1Qat, which is limited to the configuration of CBS, to allow dynamic configuration of any TSN mechanism. The amendment does not specify how the configuration should be created, but deals with how the switches interact to gather the information required for configuration generation. The amendment then defines how the configuration is made available to the switches subject to be configured. The configuration can be created according to different interaction models. Such a configuration often has to take into account the specifics of the switch model, which vary from manufacturer to manufacturer. Unfortunately, dynamic reconfiguration is in reality limited in cases such as the reconfiguration of the GCL, as the scheduling tools may not be able to create a new schedule fast enough to keep the network running when a configuration change is detected, e.g., due to a broken link. For this reason, it is common in real-time networks that both the schedule of the queues and the routing of the data frames are kept static.

5.2.2 Overview of wireless protocols

Wireless technologies can provide communication in OT environments when wiring is not an option due to mobility or cost issues, but often at the expense of poorer channel conditions. Overcoming such a lack of reliability is one of the main focuses of the technologies described in this section. In the OT field, wireless communication technologies enable additional use cases such as the rapid deployment of temporary monitoring and diagnostic networks in industrial machinery. The technologies described here are often used to establish the so-called wireless sensor networks (WSN) or wireless sensor and actuator networks (WSAN).

Bluetooth

Bluetooth, which was initially developed based on the currently unmaintained IEEE 802.15.1 standard [59], is a low-rate, short-range wireless technology primarily used to connect battery-powered consumer electronic devices. Due to its bounded medium access latency, Bluetooth has also been considered as an option in sensor and actuator networks, especially in the industrial field [60]. The communication stack covers the functionality of the seven OSI layers. In Bluetooth, speeds of up to 1 Mbps and payloads of 251 bytes at the physical-layer level are possible, which are quite limited and potentially affect scalability. The nodes in a Bluetooth network establish the so-called piconets. Piconets are star-topology networks in which the central node acts as the master and the remaining nodes, up to seven, are slaves. Such a limited number of supported devices also limits the scalability of the network. Multiple piconets that overlap spatially form a scatternet, which allows a device to participate in multiple piconets and thereby achieve a greater range in a mesh-like topology. The physical layer of Bluetooth uses FHSS, in which 79 channels are used one at a time, hopping from one to another according to a pseudo-random algorithm to avoid interference. As the technology operates in the ISM bands and considering there are overlapping piconets, interference is to be expected. An adaptive frequency hopping (FH) mechanism, which excludes channels with higher error rates, is offered in a later version of the standard.

The MAC is based on a TDMA scheme in which communication takes place via fixed-length slots. The standard proposes two modes of operation to enable data exchanges with different requirements, allowing bounded-latency and contention-free communication: asynchronous connectionless (ACL) and synchronous connection-oriented (SCO). With ACL, a master node transmits data to a set of slave nodes. The slaves respond after they have been polled by a master according to a round-robin sequence. Data under ACL is retransmitted if the master node has not received a response to the polling-based request. Retransmissions are attempted up to a certain limit before the data is considered not deliverable. However, the delay caused by retransmissions can be significant, in the order of tens of seconds. In SCO, a master communicates with a slave and receives a response in the following slot. Such slots are reserved for a specific master-slave pair and are repeated periodically, effectively protecting these slots from ACL transmissions. The presence of two methods for data exchanges partially enables the separation of data with different criticality and, to a certain extent, ensures that data from different nodes does not interfere with each other. As the protocol supports the reservation of communication resources for periodic data exchanges, it guarantees that data will be delivered unless interference occurs, enabling the fulfillment of real-time requirements. In Bluetooth networks, devices can join and leave the network dynamically, which promotes adaptability.

Apart from retransmissions in SCO, there is no specific support for fault tolerance. The protocol does not provide clock synchronization and timestamping.

Wireless Interface for Sensors and Actuators (WISA)

WISA is a wireless technology that is primarily intended to replace cables in robotic arms, as these tend to break after frequent sharp movements. Cables are removed not only for data transmission but also for power supply, with the latter being achieved after using electromagnetic field power transfer. WISA comprises a physical layer based on IEEE 802.15.1 [59] and also covers the data-link layer. WISA networks are deployed using star topologies. The technology features five channels used in parallel, which enable an aggregated data rate of up to 5 Mbps. A data exchange can comprise up to 64 bytes of physical-layer payload.

One of the five available channels is used for transmissions from the master to the slaves. The other four channels are used for transmissions from the slaves to the master, all following a coordinated frequency hopping sequence. The devices can have plain Bluetooth transceivers, while the base station, which has the role of the master, requires a special transceiver that can transmit on four channels in parallel. The channel time in WISA is divided into so-called frames, which can be used for downlink transmissions from the master to the slaves or uplink transmissions from the slaves to the master. The downlink and uplink frames are further subdivided into slots that can serve up to 120 slaves. As the technology is aimed at small transmissions, the available throughput can be considered sufficient. Therefore, the technology scales well within the given payload and throughput constraints. Since the pattern of frames and slots is repeated cyclically, the nodes have a chance to transmit data periodically, resulting in bounded-latency and contention-free transmissions, except in the case of severe interference. If a transmission is not acknowledged, it is retransmitted. The retransmission will occur at the next available opportunity, resulting in increased delivery latency. Splitting the available transmission time into slots helps to support data exchanges with different criticality and allows the channel to be split to protect transmissions coming from different nodes. The configuration step is used to associate slaves to the master and assign them the slot they have to use. The configuration can be changed during runtime [61], which has a positive effect on adaptability.

Support for data timestamping is not part of WISA.

IEEE 802.15.4

IEEE 802.15.4 [62] is a wireless standard for low-power devices that perform small data exchanges in the ISM band within short ranges of a few meters. The standard covers the physical and data-link layers. The throughput is limited to 250 kbps, with a payload at the physical layer of up to 127 bytes. With such a low throughput, the scalability of the solution is quite limited. The network nodes are divided into full function devices (FFDs) and reduced function devices (RFDs). The former can take on the role of coordinator and establish star and mesh topologies around them. RFDs are expected to have simpler hardware in order to reduce costs and energy consumption.

At the MAC level, two methods are proposed: unbeaconed and beaconed. In unbeaconed mode, the devices use CSMA/CA to send data to the coordinator, so the method cannot provide bounded-latency and contention-free communication. Data going in the opposite direction must be explicitly requested from the coordinator by the devices. In this way, a device can be inactive and save energy without having to align with the coordinator. In beaconed mode, the time is divided into cyclically repeated phases, each containing an active and an inactive segment. The active phase is subdivided into 16 fixed-length time slots, the first of which is dedicated to the beacon frame, a frame sent by the coordinator to manage the network. The non-mandatory guaranteed time slots (GTSs) feature uses the following seven slots. The slots are allocated dynamically, improving the adaptability of the technology, after an explicit request from the devices. Allocation is done via a request protocol that grants GTSs based on availability. The protocol is also used to deallocate GTSs. The allocation and deallocation of GTSs takes place during the third segment of the active phase, which is also intended for data transfers following contention-based medium access. GTSs can provide bounded-latency and contention-free communication, but since the slots are requested during the contention phase, there is no guarantee of when they will be granted. Finally, in the optional inactive phase, all nodes go into energy saving mode and do not perform any data exchanges.

The IEEE 802.15.4 standard additionally proposes the time-slotted channel hopping (TSCH) mechanism for improved reliability. The mechanism combines the use of frequency and time diversity so that transmissions can take place in different frequencies and time slots. The time slots can either be used by a single transmission so that collisions are avoided, or they can be accessed via contention. The mechanism requires the nodes to be synchronized to follow the schedule, and this happens after receiving a data or ACK frame from the time source. However, the standard leaves the assignment of frequencies and time slots open, so the fulfillment of real-time requirements is up to the scheduling algorithm.

The GTS and TSCH mechanisms offer the option of handling data with different criticality separately. By supporting time slots, they can also enable data exchanges between different nodes without them influencing each other. Retransmissions can be used for improved reliability, having two options, acknowledged, where a retransmission only occurs if the ACK is not received, and not acknowledged, where a retransmission always occurs.

IEEE 802.15.4 has recently added support for ultra-wideband which, unlike most field technologies, utilizes a large portion of the spectrum to achieve high throughput transmissions. Due to the extensive use of the spectrum, it is less affected by narrowband interference and can coexist better with licensed narrowband technologies.

WirelessHART

WirelessHART (IEC 62591 [63]) is the extension of the HART protocol to enable wireless deployments in industrial environments. It is based on the IEEE 802.15.4 physical and data-link layers, to which it adds functionality for the network, transport and application layers. The protocol can transfer payloads of up to 127 bytes at the physical-layer level at a speed of 250 kbps, which limits its scalability to low throughput applications. WirelessHART uses FHSS as an interference-avoidance mechanism. WirelessHART defines several network node roles: field device, handheld, router, adapter, access point, gateway, network manager and network security device. Field devices are the nodes with sensor and actuator functionality. Handheld are mobile nodes that are used to configure other devices. Routers are responsible for data relaying in the star- or mesh-based topology adopted by WirelessHART, and any device, including field devices, can act as a router. Adapters are used to connect wired HART devices to the WirelessHART network. Access points connect the WirelessHART network to a gateway that redirects traffic to another network. Gateways often host a network manager, which is responsible for managing communication resources, and a network security device, but separate devices can also assume both roles. End-to-end communication of industrial process data is established between the field devices themselves and the process automation controllers in the HART segment.

The MAC protocol relies on the time slots of IEEE 802.15.4, where the slots are uniquely assigned or shared between multiple participants through a contention-based access method. WirelessHART does not specify a scheduling algorithm for handling communication resources. Instead, it defines the constraints that such an algorithm shall fulfill. The constraints include the indication for reliability that data exchanges shall have two transmissions scheduled at different times and a third via a redundant path. The scheduler shall take into account that several hops may be required to reach the final destination of the data in the mesh network. ACKs are sent within the slots after the successful reception of data. The network manager node is responsible for such resource allocation and routing, along with monitoring network conditions and making adjustments to these conditions if necessary. A customized scheduling algorithm can be designed to meet the timing requirements as described in [64]. Since the data exchanges are separated in the time domain, data with different criticality or protection between data exchanges originating from different nodes is provided. Nodes can join the network dynamically, which favors adaptability, but the new configuration is not immediately available [65]. However, mechanisms to improve the timing performance of such network adaptations have already been proposed [66].

The protocol includes a method for synchronization that uses clock values that are transported as part of the ACK.

ISA100.11a

ISA100.11a [67] is an industrial wireless sensor network technology that is often depicted as the competitor of WirelessHART. Similar to WirelessHART, ISA100.11a uses the physical and data-link layers of IEEE 802.15.4, but adds the network, transport and application layers. ISA100.11a performs data transfers of up to 127 bytes of payload size at the physical-layer level at a speed of 250 kbps, which limits the scalability of the technology to very low throughput data exchanges. The devices form star or mesh topologies, and are categorized according to their functionality into I/O devices, routers, provisioning devices, backbone routers, system managers, security managers and gateways. I/O devices are assigned to sensors and actuators that provide and consume data, respectively. In the case of routers, their role is to relay data. Provisioning devices are responsible for receiving requests from other nodes to join the network. Backbone routers relay data between two networks via a backbone, with the networks and the backbone being ISA100.11a-compatible. The gateways relay data between an ISA100.11a network and a non-ISA100.11a network. The system managers take care of the network configuration, including the scheduling of network resources. Finally, the security managers are responsible for the provision of security services.

The access to the medium is based on channel hopping and takes place via uniquely-assigned time slots of configurable length. This method could potentially provide bounded-time and free-of-contention medium access. However, these guarantees depend on the implementation of a scheduling algorithm, which is left open in the standard. The hypothetical scheduler shall deal with a slot allocation that considers multi-hop transmissions. The work in [68] provides an example of a scheduling algorithm for ISA100.11a. Another medium access method offered by the protocol is slow channel hopping, where the same frequency is used over a portion of consecutive time slots. Slow channel hopping enables unbounded-time contention-based access so that devices have the chance to transmit without having to wait for their allocated time slot. Such a contention phase has the disadvantage that the network devices are required to listen to the medium for the duration of the contention phase just in case data is addressed to them, which increases energy consumption. The protocol supports retransmissions after a failed ACK. The retransmissions are performed in the next available time slot. Since ISA100.11a offers two access methods and TDMA, there is some protection against interference between data exchanges of different criticality or coming from different nodes. In addition, the allocation of time slots is configurable at runtime by the system manager, which favors the adaptability of the technology, even if the reconfiguration process is not immediate. Space redundancy is partially supported via the mesh topology, but is not a requirement as with WirelessHART.

Clock synchronization is supported by the protocol and follows a master-slave method in which a master as PAN coordinator broadcasts the reference clock value periodically via a beacon exchange and the slaves correct their values on this basis.

The protocol is highly configurable, which in contrast to the simplicity of WirelessHART can lead to interoperability problems. Nevertheless, the standard has defined profiles that take into account the specifics of different use cases in order to solve interoperability issues. One of the benefits of ISA100.11a is the implementation of a tunneling protocol that allows the encapsulation of other protocols and uses ISA100.11a for transport. In contrast, WirelessHART is confined to the use of HART.

6TiSCH

6TiSCH [69] is based on the physical and data link layers of IEEE 802.15.4. 6TiSCH uses the TSCH feature defined in the IEEE 802.15.4 standard. The IEEE 802.15.4 layers are combined with a number of IETF standards to cover the functionality described by the other OSI layers: network, transport, session, presentation and application. These IETF standards include mechanisms for secure connection to the network, scheduling management and compatibility with IPv6, which opens up the possibilities for the IoT in the industrial sector. With 6TiSCH, data transfers of up to 127 bytes of physical-layer payload with a throughput of 250 kbps can be made via star topologies, making scalability quite limited due to the low rate.

The standard includes the so-called minimal 6TiSCH profile, which defines the basic functionality for setting up a 6TiSCH-compliant network. The minimal configuration includes the definition of a TSCH schedule in which the slots are unambiguously assigned. The minimal configuration supports retransmissions via CSMA/CA. In addition to the minimal profile, the dynamic assignment of TSCH slots can be enabled. Such a process is handled via the 6top protocol (6P), which allows nodes to request the negotiation of cells, and via the Scheduling Function (SF) algorithm, which decides which cells are added to the schedule. The SF comes with a default scheduling policy, the minimal scheduling function (MSF). The MSF divides the cells into two types: The autonomous cells, which are accessed based on the node address, and the negotiated cells, which are assigned based on the nodes' requests and support the adaptability of the protocol. The MSF also includes a method for reassigning colliding cells. The support for time slots could enable separate handling of data with different criticality and protect data exchanges coming from different nodes from mutual interference. Besides the MSF, other policies can also be defined, as in [70], where a solution that considers reliability and real-time requirements can provide a contention-free access method with low latencies. 6P contains mechanisms to trigger the dynamic reconfiguration of TSCH schedules.

Clock synchronization is supported in 6TiSCH, where synchronization occurs after a node has received either a data message or an ACK message from the time provider.

ZigBee

The ZigBee protocol [71] is another technology based on the IEEE 802.15.4 physical and data-link layers. ZigBee targets low-power consumption devices, mainly sensors and actuators, over short distances. Its design is not as focused on industrial environments as other IEEE 802.15.4-based technologies such as WirelessHART or ISA100.11a, yet its features could be interesting in some cases. The protocol covers additional functionality of the network, transport, session, presentation and application layers and supports the self-organized network deployment, routing, security or device configuration. ZigBee can transfer data of up to 127 bytes of physical-layer payload with a throughput of 250 kbps, which is quite limiting for the scalability of the technology. In a ZigBee network, the devices take on the roles of end device, router or coordinator. End devices are the sensing and actuating nodes. The end devices are connected to a router, which in turn can relay data to other routers in order to set up networks with star or mesh topologies. The coordinator manages the network, starts it up and keeps it running.

The MAC protocol states that the status data of a sensor is checked by polling. Therefore, bounded-latency and contention-free communication is supported with a TDMA scheme on top of the CSMA/CA protocol. The network can be reconfigured dynamically when a device joins or leaves the network via the ZigBee coordinator, which improves adaptability. The mesh topology can favor redundancy as alternative routes can be used to cover for paths that are unavailable.

Synchronization is based on the exchange of clock information after a data exchange or ACK, as defined in IEEE 802.15.4.

Wireless Networks for Industrial Automation - Factory Automation (WIA-FA)

WIA-FA [72] is a wireless technology for factory automation. In contrast to other technologies in this area, it uses IEEE 802.11 instead of IEEE 802.15.4 for the physical and data-link layers. This enables the use of the data rates and payload offered by IEEE 802.11. The technology defines several device types: host computers, gateways, access devices, field devices and handheld devices. The devices are connected in a star topology. A gateway device connects multiple access devices, which in turn connect several field devices. Field devices can be connected to multiple access devices for redundancy.

The medium is accessed using a TDMA method with time slots that are long enough to transmit a single frame and have three different uses. Firstly, the time slots intended for the exchange of beacon frames. Secondly, the uplink time slots for transmissions from field devices to access points. Thirdly, the downlink time slots are used for transmissions from the access points to the field devices. The access to the time slots is configured dynamically by the network manager. The time slots can be allocated using three different mechanisms: scheduling, preemption and competition, covering transfers from high to low criticality. With the scheduling mechanism, bounded latency and contention free communication is possible. WIA-FA also includes the application layer, which supports industrial applications. The applications interact via three modes: client/server, publisher/subscriber and report source/sink.

WIA-FA also supports time synchronization. Fault tolerance is backed by means of a redundant star topology.

5.3 Summary of the technologies features

So far, relevant data communication technologies in the LAN and PAN area have been discussed. A person responsible for deploying a network might now wonder which technology or which combination of technologies from the ones examined could fulfill the requirements of the various use cases. Therefore, a list of attributes based on the requirements from Chapter 3 is presented below with the aim of evaluating the presented technologies based on this short list of attributes. As a relative comparison of certain aspects such as cost or complexity is difficult, these were not considered, although they are part of the requirements list. After that, Table 5.1 and Table 5.2 provide the feature summary. Please note that the information in these tables is based on the latest status at the time of publication of this thesis. However, the protocols continue to evolve and are often updated with new features, e.g., with a new physical version. Furthermore, the information collected is correct to the best of the author's knowledge and belief, but errors cannot be ruled out. Another aspect to consider is that some features are not available in some implementations of the technology or do not allow certain combinations, e.g., schedulability analysis is available but only under certain topologies. A brief explanation with a description of the possible values follows for each attribute:

- **Data size at the physical layer.** The maximum payload that the technology can transmit as provided by the physical layer. The maximum payload may not allow a strict comparison, as some technologies cover services that others do not directly include but need to be considered and may add more overhead, reducing the available payload. For this reason, the physical layer is selected as a reference. In addition, a technology may offer higher throughput even though it transmits smaller data packets. Possible values: integer in bytes.

- **Data exchange partition.** Specifies whether the technology is able to exchange data coming from different sources without the exchanges compromising each other. This is particularly important if a data source generates data with an unbounded pattern. Partitioning can be limited to a certain number of sources, e.g., two sources can coexist, but three sources may already start to affect each other. The possible values indicate whether such a partition is available: yes (Y) or no (N).

- **Data timestamping or synchronization.** Specifies whether the technology provides support for data timestamping or synchronization by default. Possible values: yes (Y) or no (N).

- **Delivery latency.** Indicates the ability of the technology to support a bounded delivery latency. A distinction based on jitter has been excluded as it is difficult to find performance data from each technology. Possible values: bounded delivery latency (B) or unbounded delivery latency (U).

- **Dynamic reconfiguration.** Specifies whether the technology can undergo reconfigurations in which data exchanges and nodes are added or removed without having to reset the communication system. The attribute refers to the adaptability requirement. Possible values: yes (Y) or no (N).

- **IT/OT.** Indicates whether the technology is classified as information or operational technology. Possible values: information technology (IT) or operational technology (OT).
- **Medium protection.** Specifies whether the transmission medium is protected against interference that could impair reliability. This attribute does not directly imply that there is a split between guided/wired or unguided/wireless medium, as wireless transmissions on a reserved frequency could also be considered protected. It refers to the requirement of reliability. Possible values: protected (P) or unprotected (U).
- **Mixed-criticality support.** Specifies whether the technology supports data with different criticality by providing separate handling, e.g., ensuring that higher-criticality traffic is not affected by lower-criticality traffic. This attribute does not exclude the mutual impact of transmissions belonging to the same criticality level, e.g., if too many transmissions of the same criticality take place at the same time. It relates to the requirements of data criticality and reliability. Possible values: yes (Y) or no (N).
- **Node interactions.** Specifies how the communication interactions between the nodes are established, whether they are restricted to certain roles or whether each node can communicate with another on an equal basis. Possible values: communication established between end systems (ES-ES), or between master and slaves (M-S).
- **OSI layers.** Specifies the OSI layers covered by the technology. Note that the technology may rely on other technologies to cover the additional functionality. However, this attribute only takes into account the layers if they are covered or referred in technology description. Possible values: 1-7.
- **Range.** Refers to the classification of the technology according to the spatial scope of its deployments. Possible values: local area network (LAN) or personal area network (PAN).
- **Redundancy.** Indicates whether and how the technology supports redundancy. Not to be confused by diversity. It relates to the requirement of reliability. Possible values: time redundancy (T), medium redundancy (M), network redundancy (N), device redundancy (D) or no redundancy (-).
- **Scheduling analysis/configuration for RT (real-time guarantees).** Indicates whether the technology requires schedulability analysis or generating scheduling configuration to provide timing guarantees. Possible values: yes (Y) or no (N).

- **Throughput.** Data rate in megabits per second. Possible values: positive real number.
- **Topology.** Network topologies allowed by the technology. Possible values: bus (B), daisy chain (D), mesh (M), ring (R), star (S) or tree (T).

5.4 Fulfillment of the requirements by the technologies

Table 5.3 and Table 5.4 summarize the extent to which each of the LAN and PAN technologies meets the requirements that the solution in this thesis is intended to cover. The reader should be aware that using a binary answer to describe whether a requirement is met is an oversimplification to ensure the readability of the table. In addition, some requirements have been excluded as they are less relevant or make a comparison between the technologies more difficult. The excluded requirements are safety, maintainability, integrity, compatibility, complexity, hardware support, and overhead and efficiency. It can be seen from the summary in the tables that none of the technologies examined alone covers the requirements that the solution in this thesis is intended to fulfill. For example, in the case of OT technologies, there is no single technology that offers the benefits of wired and wireless deployments and meets the real-time and high-throughput requirements. IT technologies, on the other hand, may offer better average throughput, but as already stated, they lack dependability and real-time requirements fulfillment.

Table 5.1: Summary of the features of LAN and PAN technologies (part 1/2)

	Data size at PHY [bytes]	Data exchange partition	Data timestamping or sync.	Delivery latency	Dynamic reconfig.	IT/OT	Medium protection	Mixed-criticality support
Ethernet	1500	N	N	U	N	IT	P	N
IEEE 802.11	2312	Y	N	B	Y	IT	U	Y
CAN	64	N	N	B	N	OT	P	Y
FlexRay	254	Y	Y	B	N	OT	P	Y
AFDX	1500	Y	N	B	N	OT	P	Y
HART	254	N	N	B	N	OT	P	N
PROFIBUS	246	Y	Y	B	N	OT	P	Y
PROFINET	1500	Y	Y	B	N	OT	P	Y
POWERLINK	1500	Y	Y	B	N	OT	P	Y
EtherCAT	1500	N	Y	B	N	OT	P	N
HSR	1500	N	N	U	Y	OT	P	N
PRP	1500	N	N	U	Y	OT	P	N
MRP	1500	N	N	U	Y	OT	P	N
TTE	1500	Y	Y	B	N	OT	P	Y
FTT-E	1500	Y	N	B	Y	OT	P	Y
TSN	1500	Y	Y	B	Y	OT	P	Y
Bluetooth	251	Y	N	B	Y	IT/OT	U	Y
WISA	64	Y	N	B	Y	OT	U	Y
IEEE 802.15.4	127	Y	Y	B	Y	IT/OT	U	Y
WirelessHART	127	Y	Y	B	Y	OT	U	Y
ISA100.11a	127	Y	Y	B	Y	OT	U	Y
6TiSCH	127	Y	Y	B	Y	OT	U	Y
ZigBee	127	Y	Y	B	Y	IT/OT	U	Y
WIA-FA	2312	Y	Y	B	Y	OT	U	Y

Table 5.2: Summary of the features of LAN and PAN technologies (part 2/2)

Node interactions	**OSI layers**	**Range**	**Redundancy**	**Sched. analysis/config. for RT**	**Throughput [Mbps]**	**Topology**	
ES-ES	1-2	LAN	N	N	10000	B, R, S, T	**Ethernet**
ES-ES	1-2	LAN	N	Y	2402	M, S	**IEEE 802.11**
ES-ES	1-2	LAN	T	Y	5	B	**CAN**
ES-ES	1-2, 7	LAN	M	Y	10	B, S	**FlexRay**
ES-ES	1-4	LAN	N	Y	1000	S	**AFDX**
M-S	1-2, 7	LAN	-	Y	0.0012	D, S	**HART**
M-S	1-2, 7	LAN	M, D	Y	12	D, R, S	**PROFIBUS**
ES-ES	1-4, 7	LAN	M, D, N	Y	100	D, R, S, T	**PROFINET**
M-S	1-4, 7	LAN	D, M	Y	1000	D, R, S, T	**POWERLINK**
M-S	1-4, 7	LAN	M	Y	100	D, R, S, T	**EtherCAT**
ES-ES	1-2	LAN	M	N	1000	R	**HSR**
ES-ES	1-2	LAN	N	N	1000	B, R, S, T	**PRP**
ES-ES	1-2	LAN	M	N	1000	R	**MRP**
ES-ES	1-2	LAN	M, D, N	Y	1000	R, S, T	**TTE**
ES-ES	1-2	LAN	D	Y	10000	S	**FTT-E**
ES-ES	1-2	LAN	M, N	Y	10000	R, S, T	**TSN**
M-S	1-7	PAN	T	Y	1	M, S	**Bluetooth**
M-S	1-2	PAN	T	Y	5	S	**WISA**
M-S	1-2	PAN	T	Y	0.25	M, S	**IEEE 802.15.4**
ES-ES	1-4, 7	PAN	T, N	Y	0.25	M, S	**WirelessHART**
ES-ES	1-4, 7	PAN	T, N	Y	0.25	M, S	**ISA100.11a**
ES-ES	1-7	LAN	T	Y	0.25	S	**6TiSCH**
M-S	1-7	PAN	N	Y	0.25	M, S	**ZigBee**
ES-ES	1-2, 7	LAN	N	Y	2402	S	**WIA-FA**

Table 5.3: Fulfillment of the requirements by LAN and PAN technologies (part 1/2). The "+" symbol means that the requirement is generally better fulfilled in the respective technology, while the "-" symbol stands for poorer coverage of the requirement.

	Data size	**Data occurrence pattern**	**Data criticality**	**Data timestamping and sync.**	**Delivery latency and jitter**	**Delivery deadline**	**Throughput**	**Range**
Ethernet	+	-	-	-	-	-	+	+
IEEE 802.11	+	+	+	-	+	+	+	+
CAN	-	+	+	-	+	+	-	+
FlexRay	-	+	+	+	+	+	-	+
AFDX	+	+	+	-	+	+	+	+
HART	-	-	-	-	+	+	-	+
PROFIBUS	-	+	+	+	+	+	-	+
PROFINET	+	+	+	+	+	+	+	+
POWERLINK	+	+	+	+	+	+	+	+
EtherCAT	+	-	-	+	+	+	+	+
HSR	+	-	-	-	-	-	+	+
PRP	+	-	-	-	-	-	+	+
MRP	+	-	-	-	-	-	+	+
TTE	+	+	+	+	+	+	+	+
FTT-E	+	+	+	-	+	+	+	+
TSN	+	+	+	+	+	+	+	+
Bluetooth	-	+	+	-	+	+	-	-
WISA	-	-	+	-	+	+	-	-
IEEE 802.15.4	-	+	+	+	+	+	-	-
WirelessHART	-	+	+	+	+	+	-	-
ISA100.11a	-	+	+	+	+	+	-	-
6TiSCH	-	+	+	+	+	+	-	+
ZigBee	-	-	+	+	+	+	-	-
WIA-FA	+	+	+	+	+	+	+	+

Table 5.4: Fulfillment of the requirements by LAN and PAN technologies (part 2/2). The “+” symbol means that the requirement is generally better fulfilled in the respective technology, while the “-” symbol stands for poorer coverage of the requirement.

Reliability	**Availability**	**Adaptability**	**Cost**	**Determinism**	**Mobility**	**Scalability**	
-	-	+	+	-	-	+	**Ethernet**
-	-	+	+	-	+	-	**IEEE 802.11**
+	+	-	-	+	-	-	**CAN**
+	+	+	-	+	-	-	**FlexRay**
+	+	-	-	+	-	+	**AFDX**
-	-	-	-	-	-	-	**HART**
+	+	-	-	+	-	-	**PROFIBUS**
+	+	-	-	+	-	+	**PROFINET**
+	+	+	-	+	-	+	**POWERLINK**
+	+	-	-	+	-	+	**EtherCAT**
+	+	-	-	+	-	-	**HSR**
+	+	-	-	+	-	+	**PRP**
+	+	-	-	+	-	-	**MRP**
+	+	-	-	+	-	+	**TTE**
+	+	+	-	+	-	+	**FTT-E**
+	+	+	-	+	-	+	**TSN**
-	-	+	+	-	+	-	**Bluetooth**
-	-	+	-	-	+	+	**WISA**
-	-	+	+	-	+	-	**IEEE 802.15.4**
+	+	+	-	+	+	-	**WirelessHART**
+	+	+	-	+	+	-	**ISA100.11a**
+	+	+	-	+	+	-	**6TiSCH**
-	-	+	+	-	+	-	**ZigBee**
+	+	+	-	+	+	+	**WIA-FA**

Part II

Contributions

Chapter 6

Overview of contributions

This thesis presents contributions grouped around three main areas:

- **Contribution area 1: detailed problem formulation, trade-offs between requirements and solutions**
  - Provides a detailed problem formulation based on a precise description of the requirements, as well as the faults and failures to be handled in data communication systems with integrated wired and wireless connectivity, where dependability and real-time requirements are in focus.
  - Reviews and evaluates the impact that satisfying a requirement could have on other requirements in data communication systems.
  - Briefly describes the envisaged solution by detailing how the requirements, faults and failures will be handled by the mechanisms proposed in contribution areas 2 and 3.
  - Addresses the research question RQ1, and partially RQ2.1 and RQ2.2.
- **Contribution area 2: fault-prevention mechanisms for the wireless medium**
  - Proposal for a wireless solution based on IEEE 802.11 to complement a wired network relying on Ethernet with TTE or TSN. The proposal uses IEEE 802.11 COTS hardware. Dependability and real-time requirements are enforced through the use of MAC-level fault-prevention mechanisms.
    - First, through a contribution that proposes a TDMA MAC protocol that coordinates access to the medium and enables differentiated handling according to timing and criticality requirements [1][2][3][7]. The differentiated handling is based on the support of TTE traffic classes or various TSN mechanisms.
    - Second, a contribution that applies frequency diversity in the communication channel using the cognitive radio approach to increase the probability of successful transmissions [5].
  - The contributions describe the scheduling problem and how it is solved.

  - The contributions are evaluated on the basis of computer simulations and a hardware implementation. The simulator and the implementation are described in detail.
  - Addresses the research question RQ2.1.

- **Contribution area 3: fault-tolerance mechanisms for wireless and wired networks**
  - Proposal of fault-tolerance mechanisms that complement the solutions from Contribution area 2.
    - One approach deals with faults in the wireless segments of the network and relies on time diversity [4].
    - Another approach handles faults that occur in the wired segment and relies on space diversity by using a wireless backup network [6].
  - The contributions describe the scheduling problem and how it is solved.
  - The contributions are evaluated on the basis of reliability analysis and computer simulations. The proposed analysis and the simulator are described in detail.
  - Addresses the research question RQ2.2.

Chapter 7

Contribution area 1: detailed problem formulation, trade-offs between requirements and solutions

7.1 Introduction

This contribution deepens the initial problem formulation and adds a precise description of the requirements, faults and failures that the solution in this thesis shall address. After clarifying the requested requirements, the trade-offs between these requirements are analyzed. Finally, the chapter enters the solution domain by exploring some of the alternatives that fulfill the requirements and justifying their selection. The solution proposal also includes a brief description of the mechanisms planned for handling faults. These mechanisms are fully elaborated in Contribution area 2 (Chapter 8) and Contribution area 3 (Chapter 9). Contribution area 1 addresses research question RQ1 and partially covers RQ2.1 and RQ2.2, which were described in Section 1.3.

7.2 Detailed problem formulation

The initial problem formulation from Section 1.1 is expanded to include details of the requirements, faults and failures that the solution in this thesis shall deal with.

7.2.1 Requirements to address

Chapter 3 has already shown how extensive and diverse the requirements for a communication system can be. However, the absolute priority is to meet the requirements for high reliability and predictable latency in critical data exchanges, as shown in [14] for industrial automation, [16] for the automotive industry, [15] for avionics and [17] for robotics. These requirements, traditionally covered by OT, may not be the only ones present in the communication system, as new requirements that serve non-critical applications and are typically covered by IT solutions could also be requested, e.g., infotainment in the automotive industry. A solution that fulfills the requirements of both critical and non-critical applications is therefore desirable to avoid scenarios where each subset of requirements is served by a different communication deployment, which would increase cost and complexity. As a result, a wider range of requirements are expected to be covered by a single solution, from critical to non-critical data exchanges, from real-time to non-real-time, covering different data sizes, occurrence patterns or different throughput, among several other dissimilar requirements. Other aspects such as mobility are often needed in industrial automation and robotics and may require the addition of wireless capabilities to the traditional wired offering for reliable and real-time deployments. Indeed, it is the wireless counterpart that is explored in more detail in this thesis, since there are comparatively fewer real-time solutions and the lower reliability of wireless deployments is still a challenge.

In the following, the entire set of requirements previously introduced in Chapter 3 are reviewed one by one to define the scope of what the proposed solution should do:

- **Data size.** The solution shall aim to support a wide range of data exchange sizes. On the one hand, the few bytes that characterize physical magnitudes that provide stimuli to real-time systems in industrial process control. On the other hand, the large data sizes typical of video streaming, be it from the low-criticality streams in video entertainment systems or the high-criticality video feeds used in computer vision cases such as autonomous driving in the automotive domain.

- **Data occurrence pattern.** The solution shall not exclude any of the described data occurrence patterns. Thus, it is expected to have applications where data exchange occurs periodically, such as data originating from a camera in computer vision cases like autonomous driving. Such applications shall coexist with sporadic events like the transmission of video streams. In addition, applications that generate unbounded events, such as an alarm that triggers a safe stop on an industrial transport line, will also need to be supported. It is assumed that periodically triggered data transmissions from both synchronized and non-synchronized data sources can be part of the use cases to be covered and the solution shall apply to both. Furthermore, supporting data exchanges with different requirements under the same infrastructure is valuable for different industries, e.g., the combination of camera sensors and control data in automotive. As mentioned earlier, it is expected that periodic and sporadic data exchanges can be guaranteed at design time, while unbounded data exchanges are subject to limited predictability.

- **Data criticality.** The solution should aim to bridge the gap between communication technologies that support critical or non-critical data exchange applications but not both. The goal is to reduce the overhead and cost of deploying multiple communication technologies by combining applications with different levels of criticality under the same communication technology without jeopardizing the requirements of critical data exchanges. By combining the handling of critical and non-critical data exchanges, use cases such as control data and infotainment in the automotive industry, can be served with a single communication solution.

- **Data timestamping and synchronization.** The solution shall be able to support the timestamping of data and the synchronization of communicating systems. This enables cases where the interpretation of the timestamp is needed, like discarding sensor data that is too old. In some cases, the communication hardware does not support hardware timestamping and the system must rely on the software. Nonetheless, the option to use hardware timestamping should still be available to those systems that can support it if they are aiming for higher precision.

- **Real-time requirement: delivery latency and jitter.** Real-time applications shall be the primary target, given their importance in scenarios such as industrial automation, the automotive industry, avionics or robotics. Therefore, different latency and jitter requirements shall be handled. Traditionally separated into different communication technologies, the proposal shall also support applications without timing requirements in the same offering, so that use cases like infotainment are also addressed.

- **Real-time requirement: delivery deadline.** Following the same reasoning from the delivery latency and jitter requirements, the solution must also meet the timing requirements taking deadlines into account.

- **Throughput.** The selected communication technology shall offer high throughput. High throughput enables applications such as streaming cameras in the automotive industry, where the amount of data is large and the exchange frequent, but also scenarios involving a large number of communicating systems, each requiring fewer data transmissions, but whose aggregated requests are considerable, like those found in industrial automation. The solution shall also provide comparable throughput levels between wired and wireless segments.

- **Range.** The applications in the fields of industrial automation, automotive, avionics or robotics that shall be handled by the solution are often in the field of LANs, where communication takes place within a radius of a few tens of meters.

- **Reliability.** The solution shall prioritize the improvement of reliability and focus on minimizing the impact of its absence, with the goal of ensuring critical data exchanges as present in industrial automation, automotive, avionics or robotics. Particular attention shall be given to the wireless counterparts in the proposed solution, as they generally perform worse than wired communication systems due to the peculiarities of the wireless channel. However, wired links can be permanently compromised if broken, thus it might be interesting to explore solutions in this regard. Due to the importance of reliability, this thesis shall also study the reliability behavior of the solution in scenarios characterized by faults of different frequency and duration. Since reliability is a priority, it is expected to trade other requirements in favor of reliability.

- **Availability.** The proposal of mechanisms for increased reliability will also have a positive impact on availability. This is an important attribute required by applications in industrial automation, automotive industry, avionics or robotics and which the solution shall have.

- **Safety.** Safety is of great relevance for some of the use cases found in industrial automation, the automotive industry, avionics or robotics. Therefore, the solution shall aim to minimize the presence of faults or to count on mechanisms to tackle them. Among the faults that may occur, criticality shall be used as a criterion to distinguish those that must be prioritized due to their potentially harmful consequences, i.e., those that have an impact on safety.

- **Maintainability.** Industrial transport lines, cars or airplanes, to name a few, go through maintenance phases. This is often due to the maintenance of their mechanical parts. In these scenarios, there are often many components, so reducing the need for maintenance for each individual component, including the communication network, is desirable. Nevertheless, this is not one of the requirements in focus.

- **Integrity.** The solution shall exchange data with integrity, as this is a precondition for a reliable data exchange. However, specific scenarios where data integrity is compromised are not the focus, as this would broaden the scope too much and deviate from the more relevant requirements of reliability and real-time. Nevertheless, the solution shall not restrict the applicability of mechanisms supporting integrity, if needed.

- **Adaptability.** Dynamic adaptation to change is a desirable characteristic of the solution, but it is not a priority and can be exchanged by other more relevant requirements, e.g., real-time guarantees. Adaptability is a requirement that strongly depends on the use case. In the automotive or avionics industry, for example, dynamic configuration is not relevant due to the relatively static deployments. In contrast, factory automation scenarios often require more flexibility as devices are sometimes added or removed and devices from different network manufacturers work together, driving the adoption of a dynamic network configuration.

- **Compatibility.** The communication technologies selected as part of the solution shall be able to complement each other so that the strengths of one technology compensate for the weaknesses of the other. For example, a wired and wireless solution benefits from the high reliability and throughput of the former and the mobility of the latter. Even though legacy systems are often present in industrial automation, automotive, avionics or robotics scenarios, interaction with these systems is often happening via gateways, which are seen just as any other node that serve as the source and destination of data transmissions. Further, the presence of legacy systems shall be taken into account due to the competition for communication resources. The solutions must therefore be able to cope with the interference caused by these legacy systems, for example.

- **Complexity.** The solution shall cover the requirements with a design that minimizes complexity the most among the options considered.

- **Cost.** The solution shall cover the specified requirements with a design that minimizes cost. For example, in the automotive industry, adding communication components in a car quickly leads to an increase in costs given the large number of vehicles produced. In airplanes, a large amount of wiring is a major problem. Examples such as the largest passenger aircraft, the Airbus A380, houses approximately 500 km of wiring [73].

- **Determinism.** The support for timely data transmissions serving highly critical applications, such as those required in industrial automation, automotive, avionics or robotics, demands a solution that minimizes uncertainty even in the presence of faults. The solution must therefore be able to fulfill the timing requirements of critical applications with sufficient reliability so that the data exchange requirements are met with high predictability.

- **Hardware support.** The selected solution shall rely on a selection of hardware that does not limit, but support the fulfillment of the other requirements, e.g., by providing a sufficiently high throughput. In addition, the solution shall preferably be based on existing COTS hardware to ease applicability and minimize costs. The ability to control the behavior of the hardware from the software, as is the case with a soft MAC, might also be advisable to enable the implementation of mechanisms that enable the fulfillment of other requirements.

- **Mobility.** The solution shall count on wireless networks' benefits, including mobility. The wireless network can also serve as a complement to wired networks, as the latter cannot fully meet mobility requirements. Mobility is particularly important in industrial automation and robotics, for example, communication devices may be installed in moving components in transport lines or robot arms. However, the case of mobile robots that move over larger areas of hundreds or thousands of square meters, e.g., mobile robots in a large warehouse, is not considered as part of the solution due to the additional complexity they might require, e.g., a handover mechanism could be needed.

- **Overhead and efficiency.** The solution shall aim to minimize the overhead of the communication protocol. For example, by defining a MAC protocol that avoids inefficient procedures like polling or token passing. However, other aspects such as support for clock synchronization are expected to increase the accounted overhead. Furthermore, solving inefficiencies due to the split of responsibilities between the communication layers are out of the scope for this thesis, but could be subject to improvements if the solutions are implemented in real projects in industrial automation, automotive, avionics or robotics. Although lower energy consumption is desired, the solution is not expected to run in an energy-constrained environment and shall not limit the fulfillment of other requirements for this reason.

- **Scalability.** The solution must be able to scale with the number of communication devices and data transmissions. The scenarios in industrial automation, avionics or robotics can range from a few to a large number of connected devices and applications. In addition, the applications communicating data may have timing requirements that must be met for an arbitrary number of connected devices and applications. Therefore, aspects such as throughput and topologies must be selected to support the scalability of the solution.

7.2.2 Faults and failures to address

In addition to reviewing the requirements and given the focus on the reliability attribute of dependability, the problem formulation is complemented by a taxonomy of expected faults and failures. Table 7.1, Table 7.2 and Table 7.3 provide a description of the faults to be handled by the proposed solution. Further, Table 7.4 and Table 7.5 contain a description of the failures to be handled by the proposed solution.

Table 7.1: Detailed problem formulation. Description of the faults to be handled by the proposed solution (1/3).

Fault 1: delayed transmission due to lack of senders coordination	
System boundaries	Internal: Lack of a MAC protocol able to coordinate transmissions.
Objective	• Non-malicious: Generally non-malicious, as it is just a consequence of the MAC protocol design. • Malicious: This may not be the most representative case, but a group of malicious nodes may try to flood the network with transmissions and deny service to legit nodes.
Intent	Non-deliberate: Non-real-time MAC protocols are designed this way because the application fields they cover often favor throughput and seamless configuration.
Persistence	Transient: Transmissions from a single device are usually not persistent and may provide the opportunity to other nodes to transmit other data.

Table 7.2: Detailed problem formulation. Description of the faults to be handled by the proposed solution (2/3).

Fault 2: lost transmission due to path loss, shadowing, multipath fading, overlapping transmissions and/or interference	
System boundaries	• Internal: Lack of a MAC protocol able to coordinate transmissions. • External: Interference from other devices transmitting simultaneously on the same channel and out of the control of the network. Also caused by path loss, shadowing and multipath fading.
Objective	• Non-malicious: Generally non-malicious, as it is just a consequence of the MAC protocol design. • Malicious: This may not be the most representative case, but a group of malicious nodes may try to flood the network with transmissions and deny service to legit nodes.
Intent	Deliberate: Non-real-time protocols are designed this way because the application domains they cover often favor throughput and seamless configuration.
Persistence	Transient: Interference is usually characterized by bursts where multiple transmissions might get affected in a row. However, interference is likely to eventually disappear and transmissions are expected to go through.

Table 7.3: Detailed problem formulation. Description of the faults to be handled by the proposed solution (3/3).

Fault 3: lost transmission due to broken wires	
System boundaries	External: A broken wire is a consequence of an external agent.
Objective	• Non-malicious: Generally non-malicious, as cables break due to physical wear. • Malicious: Broken wires could be the result of malicious intent.
Intent	• Non-deliberate: Wires usually break due to physical wear. • Deliberate: Broken wires could be the result of malicious intent.
Persistence	Permanent: Once a wire breaks, it does not repair on its own.

Table 7.4: Detailed problem formulation. Description of the failures to be handled by the proposed solution (1/2).

Failure 1: data not delivered on time	
Related faults	Fault 1: delayed transmission due to lack of sender coordination (Table 7.1)
Domain	Timing failure.
Detectability	• Signaled failure: If the transmission is expected during a time interval, it does not arrive and there is a mechanism to detect such a case. • Unsignaled failure: If the transmission is expected during a time interval, it does not arrive and there is no mechanism to detect such a case.
Consistency	Inconsistent failure: Multicast transmissions may arrive on time at some receivers while others do not receive them.
Consequences	Depends on the criticality of the data

Table 7.5: Detailed problem formulation. Description of the failures to be handled by the proposed solution (2/2)

Failure 2: data not delivered	
Related faults	• Fault 2: lost transmission due to path loss, shadowing, multipath fading, overlapping transmissions and/or interference (Table 7.2) • Fault 3: lost transmission due to broken wires (Table 7.3)
Domain	Halt failure
Detectability	Signaled failure: Signaled through lack of acknowledgment.
Consistency	Inconsistent failure: Multicast transmissions may be delivered to some receivers while others do not receive them.
Consequences	Depends on the criticality of the data

7.3 Analysis of the trade-offs between requirements

The fulfillment of requirements is generally subject to trade-offs, with requirements often conflicting and requiring to leverage the set of benefits and drawbacks from each solution. In the following, an analysis of the conflicting goals between the requirements introduced in Chapter 3 is presented. The analysis forms the basis for the solutions presented later. The analysis is not exhaustive, there are likely many other ways in which the requirements could influence each other, but describing several of these interactions and giving some examples already serves to convey an important idea: It is not possible to offer a solution that fulfills all possible requirements that can be placed on a communication system. However, a suitable solution can be found if the following is taken into account: First, relevant requirements are formulated without neglecting their trade-offs, and second, an indication is given of the minimal threshold for other, less relevant requirements, so that their fulfillment can be restricted in favor of the most relevant requirements. For example, the requirement that the communication system should simultaneously have high throughput and low energy consumption is often not possible, as high throughput usually comes at the expense of higher energy consumption. In contrast, high throughput and the ability to transmit large data sizes are more feasible together.

A summary of the impact of the requirements on each other is presented next in the form of a trade-off matrix. This is immediately followed by lists of trade-offs for each requirement, describing how the requirement interacts with the other requirements, e.g., data size vs. any other requirement. The list is not intended to be read in its entirety at once, but rather to be consulted only for the specific trade-offs in which the reader is most interested. Note that the interaction between each pair of requirements is only described once, i.e., if data size vs. throughput is described in the data size list, the same description is not found again in the throughput list. Thus, if the trade-off between a pair of requirements (e.g., throughput vs. data size) is missing in one list (e.g., the throughput list), please refer to the other possible list (e.g., the data size list).

7.3.1 Requirements trade-offs matrix

Table 7.6 and Table 7.7 provide an overview of the effects of the requirements on each other according to the descriptions provided immediately later. The tables make it clear that almost every requirement can influence all other requirements. Even in cases where such an influence is not noted in the table, it may well be that such a pair of requirements is linked in a way that is not entirely obvious.

Table 7.6: Requirements trade-offs matrix (part 1/2). An "x" in the matrix means that a trade-off has been identified.

	Data size	**Data occurrence pattern**	**Data criticality**	**Data timestamping and sync.**	**Delivery latency and jitter**	**Delivery deadline**	**Throughput**	**Range**	**Reliability**	**Availability**	**Safety**
D. size											
D. occurrence	x										
D. criticality		x									
D. time. & sync.	x	x	x								
Del. lat. & jit.	x	x	x	x							
Del. deadline	x	x	x	x	x						
Throughput	x	x		x	x	x					
Range				x	x	x	x				
Reliability	x	x	x	x	x	x	x	x			
Availability	x	x	x	x	x	x	x	x	x		
Safety		x	x	x	x	x	x	x	x	x	
Maintainability	x	x	x	x	x	x	x		x	x	x
Integrity			x	x	x	x	x		x	x	x
Adaptability	x	x	x	x	x	x	x		x	x	x
Compatibility	x	x	x	x	x	x	x	x	x	x	x
Complexity	x	x	x	x	x	x	x	x	x	x	x
Cost	x	x	x	x	x	x	x	x	x	x	x
Determinism	x	x	x	x	x	x	x	x	x	x	x
Hardware support	x	x	x	x	x	x	x	x	x	x	x
Mobility	x		x	x	x	x	x	x	x	x	x
Over. & effi.	x	x	x	x	x	x	x	x	x	x	x
Scalability	x	x	x	x	x	x	x	x	x	x	x

Table 7.7: Requirements trade-offs matrix (part 2/2). An “x” in the matrix means that a trade-off has been identified.

Maintainability	**Integrity**	**Adaptability**	**Compatibility**	**Complexity**	**Cost**	**Determinism**	**Hardware support**	**Mobility**	**Overhead and efficiency**	**Scalability**	
											D. size
											D. occurrence
											D. criticality
											D. time. & sync.
											Del. lat. & jit.
											Del. deadline
											Throughput
											Range
											Reliability
											Availability
											Safety
											Maintainability
											Integrity
											Adaptability
		x									**Compatibility**
x	x	x	x								**Complexity**
x	x	x	x	x							**Cost**
x	x	x	x	x	x						**Determinism**
x	x	x	x	x		x					**Hardware support**
		x		x	x	x	x				**Mobility**
x	x	x	x	x		x	x	x			**Over. & effi.**
		x		x	x	x	x		x		**Scalability**

7.3.2 Data characterization requirements

Data size

The following list describes how the data size requirement affects the other requirements:

- **Data occurrence pattern.** Applications require data to be sent to another computer system at a cadence specified in the data occurrence pattern. Depending on the size of the data and the transmission speed, certain occurrence patterns are not possible. For example, if the data is very large and data exchange is requested at a high frequency, there may not be enough time to finish the current transfer and start the next one. And even if the data transfer is possible, the utilization of the medium might be too high, leaving other nodes without a chance to transmit.
- **Data criticality.** No clear trade-offs identified.
- **Data timestamping and synchronization.** Sending the timestamp along with the data is expected to reduce the available payload.
- **Real-time requirement: delivery latency and jitter.** The time it takes for the data to reach the destination is proportional to the data size and is taken into account as delivery latency and jitter. If the data size changes between iterations, the latency is likely to change as well, making the jitter higher. If a specific delivery latency is required, this must be considered along with the data size and throughput requirements. For example, a large amount of data together with a low throughput technology is not expected to deliver low latencies.
- **Real-time requirement: delivery deadline.** The delivery deadline also depends on the data size, as the deadline must be set so that there is at least enough time for transmission. If a requirement is set for a specific delivery deadline, this must be taken into account together with the data size and throughput requirements. For example, a large amount of data combined with a low throughput technology may mean that the deadline cannot be met.
- **Throughput.** The combination of data size and data occurrence pattern must not exceed the throughput, otherwise the data will inevitably be discarded. In addition, other applications exchanging data in the system may require a fair share of the available throughput.
- **Range.** No clear trade-offs identified.
- **Reliability.** Data size can also have an impact on reliability. For example, some transmitted data may not be received or may be received with an error so that it has to be discarded altogether. In such a situation, more communication resources are wasted when the unfruitful data transmission is large than when it is short. Further, if the data is sent again, more resources are again required by the larger data transmission. In addition, a system with low reliability is likely to affect large data transmissions to a larger extent due to their extended use of

the communication system. Therefore, transmissions with more data are better suited for a more reliable communication technology.

- **Availability.** For systems with multiple applications, large data might cause other transmissions to wait a long time for their chance to be transmitted or even be rejected, especially if the throughput is low, which has a negative impact on availability. Depending on when and for how long the system is unavailable, large data transfers may not be possible.

- **Safety.** No clear trade-offs identified.

- **Maintainability.** Maintenance tasks could take up some of the time available for transmissions, which could limit the data size. Nevertheless, the relationship between data size and ease of maintenance is not noteworthy.

- **Integrity.** No clear trade-offs identified.

- **Adaptability.** Large data transmissions consume more communication resources, which may have a negative impact on the adaptability of the system as it is more difficult to add more traffic. The opposite case, short transmissions, can be beneficial for the adaptability of the system.

- **Compatibility.** Supporting different data sizes on different communication technologies could hinder their ability to work together and compromise their compatibility. One example for this issue is the data size differences between IEEE 802.15.4 and Ethernet.

- **Complexity.** It could be argued that a technology that requires the transmission of large amounts of data is more complex than a technology that only requires the transmission of small data portions. For example, IEEE 802.15.4 is a simpler technology than IEEE 802.11, as evidenced by the capabilities of their devices. However, it is likely possible to find counterexamples.

- **Cost.** The same explanation from the complexity requirement applies to the cost.

- **Determinism.** As previously described, when the data size varies between the transmission iterations, the delivery latency also gets affected. If the determinism is defined in such a way that the delivery jitter must be low, such changes are likely to have a negative impact on the determinism.

- **Hardware support.** Data size requirements place some demands on hardware, so technologies are often categorized depending on their support for small payloads, such as WSAN, or large payloads, such as Ethernet and IEEE 802.11.

- **Mobility.** As mobility often leads to changes in wireless channel conditions, supporting large transmissions could become more challenging. A large transmission is more easily affected by these channel variations due to its extended transmission time.

- **Overhead and efficiency.** The protocols present at each layer of the communication stack may also include some headers along with the data payload to be transmitted, a reduction in the available payload size that is considered part of the protocol overhead. When the data to be exchanged is small, the overhead from these headers is more noticeable than with larger data exchanges. Thus, the requirement to reduce overhead is generally beneficial to support larger data sizes. In addition, lower energy consumption often means that a communication system can only transmit smaller data sizes, e.g. IEEE 802.15.4.

- **Scalability.** Similarly to the case of adaptability, large data transmissions consume more communication resources, which can have a negative impact on the scalability of the system as it becomes more difficult to add more traffic.

Data occurrence pattern

The following list describes how the requirement of the data occurrence pattern affects the other requirements[1]:

- **Data criticality.** Although it is often misunderstood, the data occurrence pattern does not directly imply a specific criticality. For example, it is often assumed that an unbounded data exchange cannot be critical, but this is not always true, e.g., in the case of an alarm. However, critical data exchanges are better handled periodically or sporadically due to their predictability.

- **Data timestamping and synchronization.** Periodic synchronized data exchanges require that clock synchronization exists between the data source and the communication system that handles the data. If there is no synchronization or the quality of the synchronization is poor, the moment when data is generated and when it is processed will drift from each other.

- **Real-time requirement: delivery latency and jitter.** The delivery latency can be influenced by the data occurrence pattern. For example, periodic data exchanges could benefit from more stable latencies, as their handling is more predictable and therefore have less jitter. With aperiodic data exchanges, their associated uncertainty about the time of data generation likely translates to different transmission latencies between iterations, hence a higher jitter is expected. In addition, there may be changes between the different iterations of a recurring data exchange if, for example, the conditions on the path to the receiver end system change between iterations, e.g., due to the presence of a burst of transmissions that delays the data delivery.

- **Real-time requirement: delivery deadline.** The delivery deadline is related to the pattern of data generation, as the deadline is usually set relative to the time of data generation. This is because the validity of some data often decreases when a new instance of the data is generated. Therefore, the delivery deadline

[1] The requirements missing from the data occurrence pattern list correspond to the trade-offs described earlier in this chapter. Details can be found in the list corresponding to the other requirement of the trade-off pair.

is often set to occur at the same time or before a new instance of the data is generated. For periodic data transfers, the deadline is usually set to the exact time at which the period elapses and a new transmission is requested. Similarly, for sporadic data transmissions, a deadline could be set equal to the value of their MIT. In cases where sporadic data transmissions are characterized by a rate, there could be a burst of data transfer requests, a behavior that might make challenging to meet tight deadlines. The same reason is behind the lack of guarantees for meeting deadlines of transmissions with an unbounded occurrence pattern. As a result, the deadline of periodic data exchanges can be guaranteed at design time, since their characterization is known. Mechanisms to handle sporadic transmissions can also be part of a solution with deadline guarantees, but the unbounded generation pattern cannot be fully guaranteed.

- **Throughput.** The combination of data size and data occurrence pattern shall not exceed the throughput, otherwise the data will inevitably be discarded. In addition, other applications exchanging data in the system may request a fair share of the available throughput.

- **Reliability.** Periodic data exchanges can be guaranteed because they have an upper limit for the number of data exchanges that can be handled. Sporadic data exchanges can also be guaranteed using the same reasoning. However, if the sporadic data exchanges are specified with the rate instead of MIT, it may happen that a burst of data exchanges takes place. In such a situation, the data exchange may not be handled, which affects reliability. As for unbounded data exchanges, these cannot always be handled. It is therefore to be expected that a periodic and often sporadic data exchanges have a better reliability than unbounded data exchanges. In addition, the data occurrence pattern might be impacted negatively in systems with low reliability, as the communication resources cannot be used in certain periods and therefore frequent data exchanges are not possible.

- **Availability.** Similar to reliability, the communication system is expected to offer better availability for periodic and sporadic data exchanges, as the need for communication resources can be predicted in the worst case. However, if the sporadic data exchanges are specified with the rate instead of MIT, it may happen that a burst of data exchanges takes place. In such a situation, the communication system may not be available to perform the data exchange. With unbounded data exchanges, the worst case regarding their appearance cannot be predicted, which affects availability. Also, in systems with several applications, more frequent data exchanges might cause other transmissions to wait a long time for their chance to be transmitted, especially when throughput is low, which negatively affects availability. In addition, the data occurrence pattern might be impacted negatively in systems with low availability, as communication resources cannot be used during certain periods and therefore frequent data exchanges are not possible.

- **Safety.** Due to their better predictability, periodic and sporadic data exchanges are generally more suitable for exchanging safety-relevant data.

- **Maintainability.** Maintenance tasks may consume some of the time available for transmissions, which could also limit their occurrence. Nevertheless, the correlation between the data occurrence pattern and maintainability is not straightforward.

- **Integrity.** No clear trade-offs identified.

- **Adaptability.** The adaptability of the system could be impaired if there are existing or newly added transmissions that occur very frequently, regardless of whether this frequency is known at the design time as in the case of periodic and sporadic data exchanges, or whether it is unknown, as in the case of unbounded data exchanges.

- **Compatibility.** Supporting different data occurrence patterns in different communication technologies could affect their ability to work together and compromise their compatibility.

- **Complexity.** Dealing with different data occurrence patterns involves additional complexity, e.g., due to the design of a mechanism that allows them to coexist under the same communication system.

- **Cost.** The same explanation from the complexity requirement applies to the cost.

- **Determinism.** The effect of the data occurrence pattern on determinism depends on how determinism is defined, but it is expected that the behavior of the system can be better predicted for periodic transmissions than for sporadic transmissions. At the same time, determinism requirements could limit the supported data occurrence patterns. Determinism cannot be provided for unbounded transmissions.

- **Hardware support.** Handling data generated with different occurrence patterns might require special hardware support, e.g., to be able to perform the transfer at a certain rate. The separate handling of data generated according to different patterns in the hardware could also be beneficial. For example, a system that accommodates periodic and unbounded data exchanges in two separate queues allows periodic data exchanges not to be affected by unbounded data exchanges.

- **Mobility.** No clear trade-offs identified.

- **Overhead and efficiency.** By reducing overhead, more communication resources are available for the actual data exchanges, which could, for example, enable more frequent occurrence of data. Constraining energy consumption could limit the data occurrence patterns, e.g., a technology like IEEE 802.15.4 allows periods of inactivity. At the same time, data occurrence patterns could limit the efforts to reduce energy consumption. For example, a very frequent data exchange might not allow periods of inactivity.

- **Scalability.** Similarly to the case of adaptability, a data occurrence pattern that results in communication resources being requested so often that they are close to their maximum capacity could make it difficult to add new traffic and thus affect the scalability of the system.

Data criticality

In general, supporting the exchange of critical data could entail stricter demands towards other requirements. The following list describes how the requirement of data criticality affects the other requirements[2]:

- **Data timestamping and synchronization.** Data timestamping and synchronization might be requested by critical data transfers. Therefore, these transfers may be compromised if the quality of timestamping or synchronization is poor.
- **Real-time requirement: delivery latency and jitter.** Exchanges that observe delivery latency and jitter may also be critical. Therefore, critical transfers may be compromised if the targeted delivery latency and jitter are not met.
- **Real-time requirement: delivery deadline.** Having critical data exchanges often means stricter demands towards the delivery deadline. Further, setting a hard delivery deadline in real-time systems often implies a high criticality of the data. In contrast, a soft deadline is often an indication of lower criticality.
- **Throughput.** No clear trade-offs identified.
- **Range.** No clear trade-offs identified.
- **Reliability.** Critical data exchanges generally imply stricter demands towards reliability. For this reason, systems with low reliability are not suitable for critical data exchanges.
- **Availability.** Critical data exchanges generally implies stricter availability requirements. For this reason, systems with low availability can only support critical data exchanges to a very limited extent.
- **Safety.** Safety-relevant data exchanges often correspond to a high criticality level. A system that neglects safety is therefore unlikely to be able to support critical applications.
- **Maintainability.** Communication systems that support critical system functions might be more complicated to maintain, as special care shall be taken to avoid the negative consequences of failing to meet data exchange requirements. For this reason, it might be necessary for these critical systems to have better maintainability so that they can be easily repaired and improved.
- **Integrity.** Critical data exchanges are likely to imply stricter demands towards data integrity.

[2] The requirements missing from the data criticality list correspond to the trade-offs described earlier in this chapter. Details can be found in the list corresponding to the other requirement of the trade-off pair.

- **Adaptability.** Adapting to changes in a running system could be more challenging when supporting critical applications. Thus, if the current critical data exchanges could be affected, the system is likely to be less susceptible to change.
- **Compatibility.** Compatibility between systems or technologies can be an issue when it comes to criticality, e.g., if another system is not able to handle critical data exchanges or has a different understanding and handles criticality levels differently.
- **Complexity.** Supporting the exchange of critical data increases the complexity of the system.
- **Cost.** The same explanation from the complexity requirement applies to the cost.
- **Determinism.** Critical data exchanges generally demand stricter determinism. In addition, determinism could serve as an enabler for critical data exchanges as there is higher confidence that the critical data exchange will proceed without unexpected issues.
- **Hardware support.** Proper handling of critical data might require specialized hardware support, such as multiple queues for different levels of criticality, to store data before transmission and protect critical data exchanges from interference from less critical data exchanges.
- **Mobility.** Mobility could have an impact on critical data exchanges, e.g., through changes in the quality of the wireless channel, which could jeopardize the transmissions.
- **Overhead and efficiency.** Support for criticality could have an impact on overhead, e.g., by adding a priority to data transmissions.
- **Scalability.** Defining multiple criticality levels and handling them in the communication system could facilitate the integration of applications that exchange data of different criticality, which would have a positive effect on scalability. However, the requirements of critical applications could be more difficult to fulfill than, for example, when exchanging data with best-effort guarantees, making a communication system handling critical data exchanges less scalable.

Data timestamping and synchronization

The following list describes how the requirements for data timestamping and synchronization affect the other requirements[3]:

- **Real-time requirement: delivery latency and jitter.** Poor synchronization quality could cause communication systems to be out of sync. If synchronization is required for these systems to coordinate transmissions, it is likely that its absence would result in degradation of delivery latency and jitter.

[3] The requirements missing from the data timestamping and synchronization list correspond to the trade-offs described earlier in this chapter. Details can be found in the list corresponding to the other requirement of the trade-off pair.

- **Real-time requirement: delivery deadline.** As the delivery latency and jitter can be influenced by the synchronization quality, the same issue could affect the delivery deadline. In addition, the deadlines on the handling side are often defined relative to the time of generation. A different interpretation of the time on both sides due to poor clock synchronization quality could compromise the delivery deadline guarantees.

- **Throughput.** The use of clock synchronization protocols and timestamps as part of the data exchanges consumes some of the throughput available for the data.

- **Range.** Clock synchronization requirements might limit the range of the communication system, as the synchronization protocols usually provide less precision the larger the network becomes, e.g., due to the additional overhead caused by the intermediate nodes.

- **Reliability.** The reliability of data exchanges might be affected in scenarios where clock synchronization fails or is unable to achieve the required level of precision. This could have a significant impact on applications that require timestamps to interpret the data or on communication mechanisms that require a certain level of synchronization precision, e.g., to coordinate transmissions between computer systems.

- **Availability.** The same explanation from the reliability requirement applies to availability.

- **Safety.** The same explanation from the reliability requirement applies to safety-related data exchanges.

- **Maintainability.** It is expected that the use of synchronization will make the maintainability of the system more difficult, as it is an additional service that shall be maintained.

- **Integrity.** Meeting integrity requirements might require clock synchronization for the interpretation of data with completeness, accuracy and also to detect unauthorized modifications, e.g., via a data timestamp.

- **Adaptability.** Adaptation to changes might involve adding or removing nodes, or dealing with different channel conditions. In both situations, the synchronization of the clock can be affected. For example, when adding nodes that need to be synchronized, it might take some time for the clocks of the new nodes to synchronize with the rest of the network. In addition, the quality of clock synchronization depends on aspects like the topologies and the number of nodes.

- **Compatibility.** There may be compatibility issues between systems with different synchronization protocols or a different interpretation of such timestamping.

- **Complexity.** The use of timestamps and synchronization protocols increases the complexity of the communication solution, especially if a high precision of the synchronization protocol is required.

- **Cost.** The same explanation from the complexity requirement applies to cost.

- **Determinism.** Determinism requirements might impose stricter clock synchronization requirements in systems that rely on such a service to function. For example, if a communication system is defined as deterministic based on its fulfillment of a low data delivery jitter and if such a system uses a time-triggered scheduler to perform the transmissions, the determinism will rely on good synchronization quality. Similarly, timestamps could be used to check the validity of data using time windows, another criterion that can be used to define the scope of determinism.

- **Hardware support.** Timestamping and synchronization can be performed by different layers in the communication stack. Having the synchronization protocol at a high layer could make it easy to deploy, but is subject to the loss of precision caused by the involvement of multiple communication layers and their potential uncertainty. In some cases, a technology at a lower layer, which could be at the hardware level, might require synchronization to continue their service, e.g., to coordinate transmissions. The precision is negatively affected by the use of software timestamping, whereas hardware timestamping happens closer to the timestamp moment, allowing a better precision.

- **Mobility.** Synchronization protocols are often based on the assumption that path delays are stable, so sudden changes could affect clock synchronization algorithms. Strict clock synchronization precision requirements can affect mobility requirements, as moving nodes can cause different propagation delays.

- **Overhead and efficiency.** The use of clock synchronization protocols and timestamps introduces some overhead and reduces the available throughput. Timestamps are sent along with the data and take up some of the available payload, while clock synchronization often relies on a protocol that performs some exchanges and reduces the resources available for data transfers.

- **Scalability.** The quality of clock synchronization tends to decrease as the size of the network increases, an effect that could compromise the scalability of the network deployment.

7.3.3 Data delivery requirements

Real-time requirement: delivery latency and jitter

The following list describes how the requirements for delivery latency and jitter affect the other requirements[4]:

- **Real-time requirement: delivery deadline.** The delivery deadline shall be greater than or equal to the delivery latency, otherwise it will not be met.

[4] The requirements missing from the delivery latency and jitter list correspond to the trade-offs described earlier in this chapter. Details can be found in the list corresponding to the other requirement of the trade-off pair.

- **Throughput.** The throughput is crucial when selecting the communication technology, as it affects the timing of the data, i.e., the delivery latency and jitter. For example, low latency and tight deadlines can benefit from a technology with high throughput. If the throughput changes, which is often due to the conditions of the transmission channel, this can be expected to affect the delivery latency and cause increased jitter.

- **Range.** As the range increases, so does the propagation delay of the transmission and thus the delivery latency. In addition, the data might be transmitted via different paths and each of these paths may have a different propagation delay, which affects the delivery jitter.

- **Reliability.** In real-time systems, non-compliance with delivery latency and jitter requirements is likely to have a negative impact on reliability. In addition, reliability is also likely to impact delivery latency and jitter. For example, a common mechanism to improve reliability is to make additional transmission attempts at later times to increase the probability of a successful transmission, which affects latency.

- **Availability.** In real-time systems, missing the delivery latency and jitter requirement is expected to have a negative impact on availability. In addition, limited availability can affect delivery latency and jitter, as data exchange might be delayed until the system is available again.

- **Safety.** The lack of fulfillment of real-time requirements for safety-related data exchanges could also affect the safety of the system.

- **Maintainability.** Performing maintenance while the system is operational could be more challenging if the data exchanges have latency and jitter requirements, since these requirements shall be kept while the maintenance is ongoing.

- **Integrity.** Delivering the data with a specific latency and jitter might be required for its interpretation with completeness and accuracy, and to detect unauthorized modifications.

- **Adaptability.** Adding or removing transmissions could impact the latencies of existing transmissions. Further, new transmissions with timing requirements might not be possible due to the data exchanges already in place, e.g., support to timing requirements may require offline scheduling that does not allow for live adaptations.

- **Compatibility.** Having technologies with dissimilar guarantees for the delivery latency and jitter can lead to compatibility issues. For example, a data link layer of the TSN-based communication stack may be able to support bounded delivery latencies, while the operating system hosting the applications that send data may spoil them.

- **Complexity.** In general, systems with latency and jitter requirements can be expected to have a higher complexity as it is an additional feature that needs to be supported.

- **Cost.** The same explanation from the complexity requirement applies to the cost.

- **Determinism.** The impact of delivery latency and jitter on determinism depends on how determinism is defined. However, the ability to meet bounded delivery latencies, and in particular low delivery latency jitter values, is expected to reduce uncertainties and favor determinism in systems with real-time requirements.

- **Hardware support.** Systems with tight delivery latency and jitter requirements are expected to demand specific hardware support. For example, they might require specific hardware to be able to send data from the hardware queues with high precision. In addition, data transmission whose dispatch time is precisely controlled by the hardware often results in lower delivery jitter, in contrast to the jitter that often occurs when data is handled by software layers in the communication stack.

- **Mobility.** If mobility means a different path or a different length of path for a data exchange, this can affect the delivery latency.

- **Overhead and efficiency.** Reducing the overhead could have a positive effect on the delivery latency. For example, if the size of the headers required by the protocols in the communication is reduced, the resulting transmission is shorter and thus the latency decreases.

- **Scalability.** A scalable deployment is capable of adding more nodes or data transmissions. Such an addition is likely to require a different organization of the allocation of communication resources to transmissions, e.g., if data has to be transmitted over an additional hop, which means that the delivery latency and jitter are likely to be different.

Real-time requirement: delivery deadline

The following list describes how the delivery deadline requirement affects the other requirements[5]:

- **Throughput.** Throughput is extremely important for meeting the delivery deadline requirement, as it directly influences how fast the transmission could reach the destination. In addition, the dynamic changes in the available throughput can cause the delivery latency to vary to the extent that the delivery deadline is missed.

- **Range.** As the increased range could result in increased delivery latency, the delivery deadline also needs to be adjusted.

- **Reliability.** In real-time systems, missing the deadline requirement is expected to have a negative impact on reliability.

[5] The requirements missing from the delivery deadline list correspond to the trade-offs described earlier in this chapter. Details can be found in the list corresponding to the other requirement of the trade-off pair.

- **Availability.** The same explanation from the reliability requirement applies to availability.
- **Safety.** If the safety-relevant data exchange also has to consider delivery deadlines, the lack of their fulfillment could affect the safety of the system.
- **Maintainability.** Undergoing maintenance work while the system is in operation could be more challenging if there are data exchanges with deadline requirements.
- **Integrity.** Meeting integrity requirements could require data to be delivered before a deadline so that the data can be interpreted completely and accurately, and unauthorized modifications are also detected.
- **Adaptability.** Adding or removing transmissions may impact existing transmissions and cause them to not meet their deadlines. Further, new transmissions with deadline requirements might not be easily handled, e.g., due to the already existing data exchanges or a required offline schedule that does not allow for live adaptations.
- **Compatibility.** Having technologies with dissimilar guarantees for deadlines can be a source of compatibility issues. For example, a data link layer of the communication stack based on TSN may be able to provide bounded latencies that allow deadlines to be met, while the operating system hosting the applications that send data may spoil them.
- **Complexity.** In general, systems with deadline requirements are expected to have a higher complexity as it is an additional feature that requires support.
- **Cost.** The same explanation from the complexity requirement applies to the cost.
- **Determinism.** The impact on determinism depends on how determinism is defined. Nevertheless, in real-time systems, determinism is often based on meeting deadlines.
- **Hardware support** Systems with tight deadlines may have special requirements for hardware support. For example, they could come with a mechanism where transfers of different criticality are assigned different hardware queues so that low-criticality data exchanges do not collide with high-criticality ones and jeopardize their deadlines.
- **Mobility.** Tight delivery deadline requirements could restrict the mobility of the nodes, e.g., if such mobility requires a different path for the data to reach the destination causing different delivery latency.
- **Overhead and efficiency.** Reducing the overhead of communication can help support tighter delivery deadlines.
- **Scalability.** A scalable deployment is able to add more nodes or data transmissions, which might affect latencies and consequently deadlines

Throughput

The following list describes how the requirements for throughput affect the other requirements[6]:

- **Range.** Some low throughput technologies are also limited in range, as transmission power limits both factors. In addition, in wireless settings, the power of the transmitted signal is expected to decrease to a point at the range boundaries where more channel errors occur and thus throughput decreases.
- **Reliability.** If throughput decreases due to changes in channel conditions, some transmissions may not take place, limiting the reliability of the communication system. Furthermore, the definition of throughput implies that the data is transmitted according to the functional requirements, resulting in a clear link to reliability, i.e., reliability and throughput are proportional.
- **Availability.** The same explanation from the reliability requirement applies to availability.
- **Safety.** If a safety-relevant data exchange is no longer possible due to a reduced throughput after a change in the channel conditions, this could also affect the safety of the system.
- **Maintainability.** Due to maintenance tasks, data might not be exchanged for some time, making throughput to reach zero during these maintenance periods.
- **Integrity.** If integrity is compromised, the affected data exchanges cannot be counted as successful, causing the throughput to drop. Further, part of the throughput may be consumed by mechanisms supporting integrity, like CRCs sent along with the data transmissions.
- **Adaptability.** Sufficient throughput is beneficial for the adaptability of the system, as new transfers can be added without throughput being the limiting factor.
- **Compatibility.** Interaction between communication systems with different throughput can be a challenge, e.g., bottleneck effects could occur that affect their compatibility.
- **Complexity.** High throughput is usually the result of more complex technology, e.g., additional antennas or more advanced coding schemes.
- **Cost.** The same explanation from the complexity requirement applies to the cost.
- **Determinism.** The impact on determinism depends on how determinism is defined. Nevertheless, determinism probably requires that at least a minimum throughput is maintained so that the nodes can communicate. If the throughput

[6] The requirements missing from the throughput list correspond to the trade-offs described earlier in this chapter. Details can be found in the list corresponding to the other requirement of the trade-off pair.

varies due to variations in channel conditions, uncertainty is likely to increase and determinism will be negatively affected.

- **Hardware support.** The coding and modulation performed by the physical layer in the hardware determine the maximum throughput that the communication technology can provide.

- **Mobility.** When mobility is required, throughput can be greatly affected due to changes in channel conditions that occur depending on the location. For example, the strength of the signal decreases with the distance from the sender, resulting in a reduction in throughput.

- **Overhead and efficiency.** The channel capacity defines a theoretical upper limit for the amount of information sent over any transmission medium, i.e., the throughput. This theoretical throughput is reduced by the overhead of the communication stack. Therefore, if a certain effective throughput is required, the efficiency requirement might also need to be taken into account. In addition, higher throughput often means higher energy consumption, which has a negative impact on energy efficiency. This is the reason why battery-powered systems are often limited to offer a low throughput.

- **Scalability.** The same explanation from the adaptability requirement applies to the scalability.

Range

The following list is a description of the effects of the range requirement on the other requirements[7]:

- **Reliability.** It may happen that the reliability at the range boundaries of a wireless network is negatively affected by the decrease in signal strength. In addition, transmissions from two or more nodes within range may overlap and be lost, affecting reliability.

- **Availability.** The same explanation from the reliability requirement applies to the availability.

- **Safety.** The safety-related data exchanges could be impaired for the same reasons as explained for the reliability requirement.

- **Maintainability.** No clear trade-offs identified.

- **Integrity.** No clear trade-offs identified.

- **Adaptability.** No clear trade-offs identified.

[7] The requirements missing from the range list correspond to the trade-offs described earlier in this chapter. Details can be found in the list corresponding to the other requirement of the trade-off pair.

- **Compatibility.** Overlapping ranges between different network segments or different communication technologies in wireless settings can lead to compatibility issues. For example, two networks, one based on Bluetooth and one based on IEEE 802.11, used in the same area could interfere with each other.
- **Complexity.** A larger range can also mean that a technology involves more complexity, e.g., by adding new nodes that help to expand a network.
- **Cost.** The same explanation from the complexity requirement applies to the cost.
- **Determinism.** The impact on determinism depends on how determinism is defined. Nevertheless, determinism could be negatively affected for the same reasons as the reliability requirement.
- **Hardware support.** The range of the technology has an impact on the hardware support, e.g., different ranges could be enabled by antennas of different size or transmission power.
- **Mobility.** The maximum range defines the upper limit for the mobility of the nodes.
- **Overhead and efficiency.** A larger range could mean a larger overhead, e.g., due to the use of intermediate relaying nodes. In addition, technologies that require a longer range can be expected to consume more energy.
- **Scalability.** A scalable technology could be based on a large range that allows the network to be expanded in terms of nodes and data exchanges.

7.3.4 Dependability requirements

Reliability

The following list is a description of the effects of the reliability requirement on the other requirements[8]:

- **Availability.** An unreliable technology affects availability as the communication system is not operational when requested.
- **Safety.** Since safety requires the absence of faults that could potentially have dramatic consequences, increased reliability is expected to reduce the occurrence of such undesirable faults in safety-related data exchanges. Further, safety mechanisms might cause the communication system to enter a safe but non-functional state to protect against potentially harmful scenarios, which reduces reliability.
- **Maintainability.** Data exchanges with reliability requirements must not be impaired by maintainability tasks. For this reason, reliability requirements could make maintainability more difficult.

[8] The requirements missing from the reliability list correspond to the trade-offs described earlier in this chapter. Details can be found in the list corresponding to the other requirement of the trade-off pair.

- **Integrity.** Exchanging data with integrity is expected to be one of the requirements for reliable data exchanges.

- **Adaptability.** Reliability can be affected when new nodes or transmissions are added. A common mechanism to ensure the reliability level is to apply admission control so that the new data transmissions are rejected if they could compromise the system. In real-time systems, a schedulability analysis may be required to analyze whether the data exchanges are able to meet timing requirements. Further, some of the adjustments may not be possible at runtime. For example, it might not be possible to create a new schedule for transmissions in a short period of time, leaving out runtime adjustments.

- **Compatibility.** Technologies able to fulfill different reliability levels might be subject to compatibility issues. For example, hybrid networks based on Ethernet and IEEE 802.11 may experience issues due to the lower reliability of the wireless segments compared to the wired ones.

- **Complexity.** A reliable system often includes mechanisms that support the reliability requirement, e.g., retransmissions, which increases the complexity of the solution.

- **Cost.** The same explanation from the complexity requirement applies to the cost.

- **Determinism.** It is often assumed that a deterministic communication system should be very reliable. However, this depends on how determinism is defined. For example, the state of the communication system, e.g., whether a transmission is successful or not, might be known based on the reliability distribution over time, even if the reliability is not high. Nevertheless, it is often desirable that both features go hand in hand.

- **Hardware support.** Reliability mechanisms are often implemented in hardware, e.g., data encoding which is intended to reduce faults.

- **Mobility.** The quality of the wireless channel tends to vary depending on the location, and so does the reliability. Therefore, special care must be taken to ensure reliable data exchanges in scenarios with mobility.

- **Overhead and efficiency.** Reliability measures are often associated with additional overhead, e.g., in the case of retransmissions.

- **Scalability.** Reliability can be affected when new nodes or transmissions are added, e.g., when the communication channel is overloaded with data exchange requests adding new ones might be problematic. A common mechanism to ensure reliability is admission control, which controls the amount of data exchanges allowed in the communication system. In real-time systems, schedulability analysis is sometimes a feasible and useful tool to provide guarantees for data exchanges to meet timing requirements.

Availability

The following list is a description of the effects of the availability requirement on the other requirements[9]:

- **Safety.** Since safety requires the absence of faults that could potentially have dramatic consequences, the fact that the system is available implies no such faults are present that will compromise the safety-related data exchanges. Further, safety mechanisms might cause the communication system to enter a safe but non-functional state to safeguard against potentially harmful scenarios, thereby reducing availability.

- **Maintainability.** For systems that require high availability, maintenance could become more challenging due to the potential disruptions maintenance could cause in the service.

- **Integrity.** Exchanging data with integrity is expected to be one of the requirements for available data exchanges. Thus, if the system is available, this means that the data exchanges are performed observing integrity.

- **Adaptability.** Availability can be affected when new nodes or transmissions are added, as communication resources are limited and may not be sufficient to meet the additional transmission requirements. To ensure that the data exchanges can be fulfilled according to the requirements, it is common to apply admission control. If the requirement to add new data exchanges cannot be met, the admission control mechanism rejects them, which has a negative impact on the availability of the communication system. The admission control mechanisms might require a schedulability analysis to decide whether the data exchanges with real-time requirements can be fulfilled. In addition, some of the adjustments may not be possible at runtime due to the risk of affecting the existing data exchanges, e.g., if a new schedule for transmissions cannot be obtained in a short time and thus adaptations are not possible.

- **Compatibility.** Problems can arise with technologies that fulfill different availability levels. In hybrid networks based on Ethernet and IEEE 802.11, for example, problems may be experienced to the generally lower availability of the wireless segments compared to wired ones.

- **Complexity.** A more available system often counts with mechanisms that support such a requirement, e.g., mechanisms for improved reliability, which increases the complexity of the solution.

- **Cost.** The same explanation from the complexity requirement applies to the cost.

- **Determinism.** It is often assumed that a deterministic communication system should have high availability. However, it depends on how determinism is defined.

[9] The requirements missing from the availability list correspond to the trade-offs described earlier in this chapter. Details can be found in the list corresponding to the other requirement of the trade-off pair.

Nevertheless, it is often desirable that both features go hand in hand. It is also important to characterize the outage phases in order to reduce uncertainties that harm determinism.

- **Hardware support.** Mechanisms that support the availability of the system are often implemented in hardware, e.g., through backup hardware that enables a secondary path for data transmissions in case the primary path is faulty.
- **Mobility.** Availability requirements might limit mobility requirement as the quality of the wireless channel varies by location.
- **Overhead and efficiency.** Implementing mechanisms for additional availability can lead to overhead, e.g., when implementing retransmissions.
- **Scalability.** Availability can be affected when new nodes or retransmissions are added, e.g., when the communication channel is overloaded with data exchange requests. To ensure that the data exchanges can be performed according to the requirements, it is common to apply admission control. If the request to add new data exchanges cannot be met, the admission control mechanism rejects them, which has a negative impact on the availability of the communication system. The admission control mechanisms might require a schedulability analysis to decide whether the data exchanges with real-time requirements can be fulfilled.

Safety

The following list is a description of the effects of the safety requirement on the other requirements[10]:

- **Maintainability.** Communication systems that support safety-related system functions might be more complicated to maintain, as special care must be taken to avoid the negative consequences of failing to meet safety-relevant data exchange requirements.
- **Integrity.** Having safety-relevant data will probably entail stricter data integrity requirements for these exchanges.
- **Adaptability.** Adapting to changes in a running system can be more challenging when supporting safety-relevant applications, as these changes can have a negative impact on safety, for example if new data exchanges overload the channel. Thus, if the current safety-relevant data exchanges could be compromised, the system is likely to be less susceptible to change.
- **Compatibility.** The compatibility between technologies can be a problem when it comes to safety, e.g., technologies with different levels of safety fulfillment could have problems working together.
- **Complexity.** It is expected that the support for safety-relevant data exchanges will increase the complexity of the system.

[10] The requirements missing from the safety list correspond to the trade-offs described earlier in this chapter. Details can be found in the list corresponding to the other requirement of the trade-off pair.

- **Cost.** The same explanation from the complexity requirement applies to the cost.
- **Determinism.** The impact on determinism depends on how determinism is defined. However, having safety-relevant data exchanges generally implies stricter requirements for determinism. In this type of data exchanges, having safety measures that define the outcome of failures has a positive effect on knowing what will happen, i.e., they bring certainty. Hence, the communication system will either perform the data exchange according to the requirements or be able to transition to a safe state in the event of an error.
- **Hardware support.** To ensure that safety is handled correctly, special hardware support may be required, e.g., by having a cyclic redundancy check (CRC) mechanism to support integrity in the hardware.
- **Mobility.** Safety-relevant data exchanges can be affected by changes in the quality of the wireless channel that may occur due to mobility.
- **Overhead and efficiency.** Supporting safety mechanisms could have an impact on overhead, e.g., by adding a CRC that consumes part of the available payload.
- **Scalability.** The requirements of safety-relevant applications are generally more difficult to fulfill than e.g., best-effort data exchanges, making a communication system that exchanges safety-relevant data likely less scalable.

Maintainability

The following list describes how the maintainability requirement affects the other requirements[11]:

- **Integrity.** No clear trade-offs identified.
- **Adaptability.** No clear trade-offs identified.
- **Compatibility.** No clear trade-offs identified.
- **Complexity.** A system that requires ease of maintenance is associated with additional complexity.
- **Cost.** The same explanation from the complexity requirement applies to the cost.
- **Determinism.** The impact on determinism depends on how determinism is defined. Nevertheless, the maintenance of the system could have a negative effects on determinism if it is not a predictable process and the system is not able to handle to such unpredictability.

[11] The requirements missing from the maintainability list correspond to the trade-offs described earlier in this chapter. Details can be found in the list corresponding to the other requirement of the trade-off pair.

- **Hardware support.** A system that requires maintainability might be supported by specialized hardware, e.g., by having a backup hardware that keeps the system running while maintenance is performed.

- **Mobility.** No clear trade-offs identified.

- **Overhead and efficiency.** Maintenance tasks might consume part of the resources of the communication system, e.g., part of the computation capabilities of the communication devices or part of the time available for transmissions, which increases the overhead.

- **Scalability.** No clear trade-offs identified.

Integrity

The following list is a description of the effects of the integrity requirement on the other requirements[12]:

- **Adaptability.** No clear trade-offs identified.

- **Compatibility.** No clear trade-offs identified.

- **Complexity.** Adding mechanisms to support integrity is likely to increase the complexity of the system.

- **Cost.** The same explanation from the complexity requirement applies to the cost.

- **Determinism.** The impact on determinism depends on how determinism is defined. However, guaranteeing that the exchanged data is complete, accurate and has not been altered is expected to reduce uncertainty and back determinism.

- **Hardware support.** The mechanisms to support data integrity might need to be handled with special hardware support, e.g., by implementing CRC in hardware.

- **Mobility.** No clear trade-offs identified.

- **Overhead and efficiency.** The use of mechanisms to support data integrity could add some overhead, e.g., the use of a CRC.

- **Scalability.** No clear trade-offs identified.

[12] The requirements missing from the integrity list correspond to the trade-offs described earlier in this chapter. Details can be found in the list corresponding to the other requirement of the trade-off pair.

7.3.5 Other requirements

Adaptability

The following list describes how the requirement of adaptability affects the other requirements[13]:

- **Compatibility.** An adaptive technology could be helpful to solve compatibility issues. For example, the two technologies IEEE 802.15.4 and IEEE 802.11 use overlapping transmission frequencies, and the adaptations in the use of the medium access make it possible to avoid collisions between the two systems.

- **Complexity.** The need to adapt to changes is likely adding some complexity to the system.

- **Cost.** The same explanation from the complexity requirement applies to the cost.

- **Determinism.** An adaptable system is expected to react to changes. These changes put the system in different states that must be taken into account for the system to be deterministic. In contrast, the requirements for determinism could mean that the system is less susceptible to change, which has a negative impact on its adaptability.

- **Hardware support.** Adaptation to changes might depend on hardware support, e.g., being able to switch the transmission frequency to avoid an interfered channel. However, implementing parts of the communication stack in hardware rather than software could limit adaptability. For example, non-SDR implementations based entirely on hardware generally offer less adaptability.

- **Mobility.** The changes that the system needs to adapt to could also impact mobility, for example if the addition of a new node allows for greater range. In addition, an adaptable communication system can be beneficial when mobility is required, e.g., if the system can adapt to the varying channel conditions at the different locations.

- **Overhead and efficiency.** Supporting adaptation to changes, e.g., by implementing an admission control system, is an additional mechanism that is likely to cause some overhead in communication.

- **Scalability.** Having a scalable technology could be a prerequisite for supporting adaptations, especially if the adaptation requires adding nodes or data exchanges.

[13] The requirements missing from the adaptability list correspond to the trade-offs described earlier in this chapter. Details can be found in the list corresponding to the other requirement of the trade-off pair.

Compatibility

Compatibility requirements can impact all other communication requirements, as compatibility evaluates whether the combination of two technologies enables the fulfillment of the other requirements. For example, if a high throughput technology is connected to another low throughput technology, the solution might need to be adapted to the bottleneck caused by the low-throughput technology in order to improve their compatibility. The following list describes how the compatibility requirement affects the other requirements[14]:

- **Complexity.** The combination of technologies and the resolution of potential compatibility issues is likely to lead to increased complexity.
- **Cost.** The same explanation from the complexity requirement applies to the cost.
- **Determinism.** When demanding determinism from a communication system, it shall be ensured that these technologies are compatible in every aspect that might affect the determinism. For example, in a deterministic system that relies on clock synchronization, multiple nodes may provide dissimilar levels of precision for the clock synchronization, which might degrade the quality of clock synchronization and compromise determinism.
- **Hardware support.** Compatibility issues often occur at the hardware level. For this reason, there are certification organizations such as the Wi-Fi Alliance, which ensure that IEEE 802.11 devices are compatible across manufacturers.
- **Mobility.** No clear trade-offs identified.
- **Overhead and efficiency.** Combining technologies that were designed independently, even if they are intended to work together, is likely to come at the price of lower efficiency. For example, different layers in a node's communication stack might make copies of the data to be transmitted instead of processing the data from a single memory location, causing additional overhead in the memory operations.
- **Scalability.** No clear trade-offs identified.

Complexity

The goal of reducing the complexity of the communication system may affect any of the other requirements, since meeting the requirements is likely to introduce more complexity into the system. However, there may be different ways in which a requirement can be satisfied and each of them may have a different complexity. The impact of the complexity requirement on the other requirements[15] is described below:

[14] The requirements missing from the compatibility list correspond to the trade-offs described earlier in this chapter. Details can be found in the list corresponding to the other requirement of the trade-off pair.

[15] The requirements missing from the complexity list correspond to the trade-offs described earlier in this chapter. Details can be found in the list corresponding to the other requirement of the trade-off pair.

- **Cost.** Complexity often translates to higher costs.
- **Determinism.** Requiring the system to be deterministic could be complex to fulfill, e.g., by requiring the generation of time-triggered schedules. In contrast, determinism may also help to limit the number of states and transitions in the system, which in turn can lead to simpler systems.
- **Hardware support.** There are often trade-offs to consider between managing complexity in software or in hardware, e.g., SDR vs non-SDR systems.
- **Mobility.** Requiring to handle mobility, e.g., supporting handover processes or the ability to adapt to changing channel conditions, is likely to increase the complexity of the solution.
- **Overhead and efficiency.** Reducing overhead and striving for an efficient solution pose challenges that can bring additional complexity.
- **Scalability.** A technology that exhibits scalability could have additional complexity as it has to handle use cases with a different number of nodes and data exchanges.

Cost

Having cost limitation requirements can have a negative impact on other requirements. Such a negative impact often means a less advanced solution with fewer features. The following list describes how the cost requirement affects the other requirements[16]:

- **Determinism.** The requirement for the system to be deterministic is likely to be more complex to fulfill, e.g., due to the need to create time-triggered schedules, compared to supporting best-effort data exchanges, which typically have more relaxed requirements. This increased complexity is also likely to result in higher costs.
- **Hardware support.** No clear trade-offs identified.
- **Mobility.** Requiring to handle mobility, e.g., support for handover processes or the ability to adapt to changing channel conditions, is likely to increase the complexity of the solution and therefore increase costs.
- **Overhead and efficiency.** No clear trade-offs identified.
- **Scalability.** A technology that exhibits scalability could come with additional complexity as it has to handle use cases with a different number of nodes and data exchanges. This additional complexity is likely to be reflected in higher costs.

[16] The requirements missing from the cost list correspond to the trade-offs described earlier in this chapter. Details can be found in the list corresponding to the other requirement of the trade-off pair.

Determinism

In general, the effect of determinism on other functional requirements depends on the terms used to define determinism, i.e., whether a particular system state is part of the expected. The following list describes how the determinism requirement affects the other requirements[17]:

- **Hardware support.** Determinism can be backed by mechanisms supported in the hardware, e.g., multiple hardware queues to separate periodic and unbounded data transmissions, so that the periodic transmissions can be handled in a deterministic manner.
- **Mobility.** Changes in the wireless channel conditions due to mobility might compromise determinism.
- **Overhead and efficiency.** Some overhead could be associated with mechanisms designed to support determinism. For example, a deterministic assignment of time-slots could be enforced by polling or by time-triggered schedules backed by a clock synchronization protocol, in both cases incurring additional overhead.
- **Scalability.** Depending on the terms used to define determinism in a communication system, scalability may be affected. For example, if determinism requires very low delivery jitter in each data exchange, it may not be possible to add new nodes or data exchanges to the network that could affect such jitter.

Hardware support

Hardware support is often an enabler for functional requirements. However, certain functionalities are more typical to be covered in hardware and some others in software. The following list describes how hardware support affects the other requirements[18]:

- **Mobility.** Hardware support is required for the implementation of wireless communications, which is the primary enabler of mobility.
- **Overhead and efficiency** The goal of reducing overhead or improving efficiency can affect the design and selection of hardware. For example, the hardware that implements the PCF could be discarded in favor of other alternatives given the overhead of the PCF polling protocol.
- **Scalability.** Hardware support is behind some of the requirements for scalability, such as high throughput or greater range.

[17] The requirements missing from the determinism list correspond to the trade-offs described earlier in this chapter. Details can be found in the list corresponding to the other requirement of the trade-off pair.

[18] The requirements missing from the hardware support list correspond to the trade-offs described earlier in this chapter. Details can be found in the list corresponding to the other requirement of the trade-off pair.

Mobility

The effects of the mobility requirement on the other requirements[19] are described below:

- **Overhead and efficiency.** Handling mobility could cause some overhead, e.g., to manage dynamic topology changes.
- **Scalability.** No clear trade-offs identified.

Overhead and efficiency

Overhead and efficiency requirements can have an impact on any other requirement. The description of the effects of overhead and efficiency requirement on the other requirements[20] follows below:

- **Scalability.** Scalability requires that sufficient communication resources are available so that new nodes or data transmissions can be added. To support scalability, it may therefore be necessary to reduce the overhead.

Scalability

As with adaptability, the system is defined as scalable based on the requirements it must fulfill, e.g., a system that supports real-time and critical data exchanges is scalable if new nodes or applications requesting data exchanges can be added without compromising real-time and criticality properties. In the case of scalability, which is the last requirement in the list, all trade-offs between scalability and other requirements have already been described earlier in this chapter. Details can be found in the list corresponding to the other requirement of the trade-off pair.

7.4 Solution proposal

A proposal for a solution to the problem previously detailed in Section 7.2 is outlined by describing how the requirements and faults are to be handled. The solution proposal ends with a summary of the committed solution.

7.4.1 Addressing the requirements

The review of data communication requirements conducted in Chapter 3 and the subsequent analysis of their trade-offs in Section 7.3 has shown how complicated could be to meet requirements that depend on each other and might be contradicting. Further,

[19] The requirements missing from the mobility list correspond to the trade-offs described earlier in this chapter. Details can be found in the list corresponding to the other requirement of the trade-off pair.

[20] The requirements missing from the overhead and efficiency list correspond to the trade-offs described earlier in this chapter. Details can be found in the list corresponding to the other requirement of the trade-off pair.

a side-by-side comparison of the requirements and coverage by the different technologies in Chapter 5 also showed that a single technology may not be able to meet all the requirements of a set of use cases.

Next, the requirements that the solution shall address are revisited. For each requirement, it is discussed which technologies come into question and which fulfill them best. The aim is to create a shortlist of technologies to be used in the solution:

- **Data size.** Ethernet and IEEE 802.11 are capable of transmitting data from a few bytes to several hundred bytes, enabling a wide range of use cases, including industrial process control and video streaming. Therefore, Ethernet and IEEE 802.11 are selected as part of the proposed solution. In contrast, typical industrial and automotive buses offer smaller payloads that cannot handle large data transfers. Similarly, IEEE 802.15.1- and IEEE 802.15.4-based technologies offer solutions for short data transmissions and are therefore discarded.

- **Data occurrence pattern.** Most of the listed data communication technologies are capable of transmitting data regardless of whether the applications generate them with periodic, sporadic or unbounded patterns. One exception is HART, where the application layer is part of the outlined communication stack and does not allow data to be generated in patterns other than periodic. Some of the technologies allow data to be handled differently depending on the generation pattern, which translates into improved performance in aspects such as latency or reliability. This is the case for protocols with separate phases for periodic and aperiodic data transmissions such as IEEE 802.11, FlexRay, POWERLINK, TTE, FTT-E, TSN, IEEE 802.15.4 or ISA100.11a. These data handling aspects, often covered by the MAC layer, are frequently not fully defined in the technology and leave loose ends such as the scheduling algorithm to the user. In addition, the MAC protocols offered by these technologies are usually subject to performance improvements, e.g., by removing inefficient mechanisms like polling. The proposed solution enables different treatment of data generated with different patterns by means of a TDMA-based wireless MAC protocol using the TTE traffic classes or different TSN mechanisms. This separate treatment enables a mix of use cases, including handling periodically generated data as provided by sensors or sporadically generated data as in video streaming, both with time guarantees. Furthermore, the proposal extends the scheduling solution for TTE and TSN to enable such support in combined wired and wireless scenarios.

- **Data criticality.** Technologies like CAN, FlexRay, PROFIBUS, PROFINET, POWERLINK, TTE, FTT-E, TSN, Bluetooth, WISA, IEEE 802.15.4, WirelessHART, ISA100.11a, 6TiSCH, ZigBee or WIA-FA provide means to differentiate data into two or more criticality levels. Such differentiation can be done using priorities, assigning data to different traffic classes or enabling a different medium access protocol for them, e.g., contention-free vs. contention-based. In this way, these technologies guarantee that critical traffic is not affected by non-critical traffic. The solution proposed in this thesis utilizes the TTE traffic classes or different TSN mechanisms to enable such differentiated treatment. This support facilitates critical and non-critical use cases like control data and infotainment, respectively, in the automotive domain. Furthermore, the proposal extends the scheduling solution for TTE and TSN to enable such support in combined wired and wireless scenarios.

- **Data timestamping and synchronization.** In the proposed solution, synchronization between nodes is provided based on PTP, as the performance of PTP is proven in Ethernet networks and the software is available for different operating systems. Having clock synchronization available enables use cases where interpretation of timestamps originating from different devices is needed, and also supports some of the TTE traffic classes and TSN mechanisms that would not work without clock synchronization. The performance of the protocol over the wireless medium is one of the aspects studied, along with finding the configuration that achieves the best result. Such an evaluation is useful to know the limitations of clock synchronization protocols that do not provide hardware support and run as any other software on a general-purpose operating system.

- **Real-time requirement: delivery latency and jitter.** The support to data having different timing requirements, from guaranteed latency to no guarantees, as part of the same communication technology is provided in IEEE 802.11, FlexRay, PROFIBUS, PROFINET, POWERLINK, TTE, FTT-E, TSN, IEEE 802.15.4, WirelessHART, ISA100.11a, 6TiSCH, ZigBee and WIA-FA. These technologies enable scheduled collision-free transmissions for data with guaranteed latency. Further, the technologies offer unscheduled or subject to collision transmissions for data exchange without guarantees. Both alternatives are categorized in this thesis into traffic classes similar to those of TTE or the different traffic handling guarantees provided by the TSN mechanisms. This support enables relevant real-time applications such as those required in industrial automation, the automotive industry, avionics or robotics. In addition, these mechanisms also enable data exchanges without timing guarantees, e.g., in infotainment. Furthermore, the proposal extends the scheduling solution for TTE and TSN to enable such support in combined wired and wireless scenarios.

- **Real-time requirement: delivery deadline.** Following the same reasoning from the delivery latency and jitter requirements, the timing requirements with respect to deadlines are met by supporting traffic classes similar to those of TTE or by the handling mechanisms of TSN. The scheduling solution is extended accordingly to enable such support in combined wired and wireless scenarios.

- **Throughput.** The proposed solution is based on Ethernet and IEEE 802.11, as these support high throughput. Legacy buses used in industrial and automotive applications offer a throughput that is not capable of handling large data transfers. Similarly, IEEE 802.15.1- and IEEE 802.15.4-based technologies offer low throughput and are therefore discarded.

- **Range.** Technologies from the PAN are left out of the selected solution in favor of the LAN-based technologies Ethernet and IEEE 802.11 due to their more suitable range. The performance of LANs has proven successful in home and office environments. This thesis explores how LANs could also be applied for communication in scenarios in other fields but with ranges of the same magnitude, such as in industrial factories, cars or airplanes. The support to the handover process is not foreseen, but solutions could be explored in the future if needed.

- **Reliability.** A whole set of mechanisms is proposed in this thesis whose main goal is to ensure the critical data exchanges that are common in industrial automation, automotive, avionics or robotics scenarios. First of all, the solution includes mechanisms to detect duplicate, lost, delayed, out-of-sequence or corrupted data. The mechanisms could be based on acknowledgments, the detection of discrepancies between scheduled and actual communication, or CRC, among others. In the wired segment, a technology such as TTE or TSN can guarantee a high level of reliability. However, even if wired networks are often more reliable than wireless networks, transmissions still rely on physical connections over wires that can sometimes be broken, leading to permanent failure and compromising reliability. To cover this case, reliability improvements are explored in this thesis by proposing space diversity mechanisms that combine a wired network with a wireless backup channel. In the wireless segment, the selected IEEE 802.11 is more limited by the negative consequences of the lack of coordination of medium access, fading, shadowing and interference. Therefore, the proposed solution aims to close this gap by introducing a mixture of fault-prevention and fault-tolerance mechanisms. First, the TDMA-based MAC protocol is proposed to avoid overlapping transmissions. The support for different reliability levels is based on different traffic classes similar to those of TTE or corresponding to the different handling mechanisms of TSN. An additional fault-prevention mechanism is proposed that relies on cognitive radio. Second, considering that fading, shadowing or interference may still be present, mechanisms against these factors are proposed in this thesis. These mechanisms that counteract the channel issues are based on diversity techniques that exploit time diversity.

- **Availability.** The proposed mechanisms to enable high reliability should have a similar positive impact on availability and support the availability requirements of applications in industrial automation, the automotive industry, avionics or robotics.

- **Safety.** The proposed solution supports the safety requirement by providing a reliable technology in which faults are largely avoided and, if they do occur, are tolerated so that they do not pose a risk. However, some faults can still occur and lead to failures. More advanced mechanisms that react to such failures and trigger safety reactions are out of the scope of the solution. Nevertheless, these should be able to be handled by upper-layer protocols, as the proposed communication stack is flexible. For example, the openSAFETY protocol can detect such failures independently of the underlying technologies used to perform the data transmission [74].

- **Maintainability.** This requirement is not the focus of the proposed solution. However, maintainability can benefit from the introduction of robust technologies based on the well-known Ethernet and IEEE 802.11 standards, as these technologies are implemented in COTS hardware that has been proven on a large scale.

- **Integrity.** Specific scenarios in which data integrity is at risk are not the focus of the proposed solution. However, the solution is not intended to limit the applicability of data integrity protection measures when needed for a specific use case. For example, upper layer protocols could be included in the communication stack to cover this purpose, if needed.

- **Adaptability.** There are numerous examples of technologies that can reconfigure their medium access scheme to adapt to changing traffic requirements without affecting their support to data transfers with timing constraints. For example, IEEE 802.11, FlexRay, POWERLINK, FTT-E, TSN, Bluetooth, WISA, IEEE 802.15.4, WirelessHART, ISA100.11a, 6TiSCH, ZigBee or WIA-FA. Other technologies such as Ethernet and IEEE 802.11 have medium access methods that do not offer time guarantees, but would be able to respond to changes in data transmission demands due to their decentralized protocol that requires little coordination. Nevertheless, the adaptability in the proposed solution is mainly focused on less critical data transfers. Support for reconfiguration of highly critical data transfers could be possible if needed, subject to preconfigured schedules or, less practicably, sufficiently fast generation of schedules that adapt to the new data transmission requirements.

- **Compatibility.** The proposed solution relies on wired and wireless communication technologies to leverage the strengths of both. Therefore, the design should favor their interoperability and enable data transmissions across both segments. Existing examples of such tandems are HART/WirelessHART and Ethernet/IEEE 802.11, the latter being chosen because it best fulfills other requirements. In addition, the solution relies on the traffic classes of TTE or TSN mechanisms, which are primarily designed for Ethernet, but are extended to IEEE 802.11 in the proposed solution. Other technologies, such as PTP for clock synchronization, are part of the solution, as PTP is often found in Ethernet deployments. The proposed solution does not take into account legacy systems, apart from the fact that they might be located nearby and could become a source of interference – a problem that is addressed with fault-prevention and fault-tolerance mechanisms.

- **Complexity.** Meeting the heterogeneous requirements that the solution shall cover leads to increased complexity. Nevertheless, the aim is to select the less complex solution that fulfills the requirements. In addition, the use of COTS solutions can help to reduce complexity, as the proposal only needs to build on the existing and proven physical- and link-layer mechanisms.

- **Cost.** The use of IEEE 802.11 COTS should reduce development and deployment costs, even if some enhancements to the base solution are required to meet the demanded requirements. It is also a common approach that applications with different requirements use different communication networks. However, the proposed solution combines the fulfillment of distinct application requirements over a single technology for the wired segment, Ethernet, and a single technology for the wireless segment, IEEE 802.11. By complementing wired networks with wireless capabilities, the often higher cost of a wired deployment alone is likely reduced.

- **Determinism.** Determinism is supported by mechanisms for fault prevention and fault tolerance. The proposed solution discards technologies with unbounded or contention-based medium access for critical data exchanges due to their low reliability. Therefore, bounded and contention-free medium access for critical data exchanges in combination with scheduled communication resources is part of the solution preventing faults. The use of cognitive radio is also meant to reduce faults. In addition, time-triggered resource allocation supports determinism in complex scenarios, including multi-hop networks with different physical layers and a mix of timing requirements. The fault-tolerance mechanisms based on time- and space-diversity can help maintain determinism even in the presence of faults. Characterizing the behavior of the proposed mechanisms in different scenarios, focusing on the problems of interference and broken wires, will help to evaluate the extent to which determinism is maintained.

- **Hardware support.** The proposed solution utilizes widely available IEEE 802.11 COTS hardware, and considers Ethernet and IEEE 802.11 as the natural counterparts. The choice of hardware is also in line with the latest trends towards the adoption of TSN, which is almost exclusively applied to Ethernet and is taken over to IEEE 802.11 in this proposal. The selected hardware is also an enabler for other requirements, such as support for different data sizes, high throughput or LAN range. The selected IEEE 802.11 hardware is based on the soft MAC approach, which allows more flexibility in implementing the additional mechanisms needed to fulfill the requirements.

- **Mobility.** This requirement, which is frequently encountered in industrial automation and robotics, is one of the main reasons for the introduction of a wireless counterpart alongside the wired network.

- **Overhead and efficiency.** The guarantee of timing requirements is often based on mechanisms that cause considerable overhead, such as token passing or polling, as found in IEEE 802.11, PROFIBUS, POWERLINK or ZigBee. Mechanisms based on pre-assigned slots, such as those found in FlexRay, PROFINET, TTE, FTT-E, TSN, WISA, IEEE 802.15.4, WirelessHART, ISA100.11a or 6TiSCH, cause less overhead than token passing or polling. Sometimes the overhead is limited by allowing the polling message or token to transmit data, such as with HART, EtherCAT or Bluetooth. However, this comes at the price of a less flexible data exchange mechanism that does not work well if the transmissions between the different sending nodes are not balanced. The goal of the proposed solution is to use a pre-assigned slots mechanism to indicate which node has a chance to transmit at a given time without causing much overhead. Furthermore, the PTP-based synchronization protocol introduces some overhead. A disadvantage of the selected solution is that Ethernet and IEEE 802.11 cause significant protocol overhead for small data transmissions. The solution is not aimed at energy-constrained environments, mainly because of the negative impact it would have on throughput.

- **Scalability.** Network deployments based on a switched-Ethernet topology often scale well with the number of nodes and data transmissions. In wireless environments, a shared collision domain limits the number of data transmissions, as simultaneous transmissions on the same frequency are not possible. In addition, Ethernet and IEEE 802.11 come with high throughput, which has a positive effect on scalability. Scalability in the solution is thus supported by bounded- and contention-free medium access, which sorts the use of the channel among several users, and high throughput.

7.4.2 Addressing the faults

In this section, a number of fault-handling techniques addressing the faults that were described in Section 7.2.2 are proposed. They are briefly described in Table 7.8, Table 7.9 and Table 7.10. The failures are not explicitly addressed, as the treatment of the faults is meant to reduce the failures.

Table 7.8: Description of the fault handling by the proposed solution (1/3).

Fault 1: delayed transmission due to lack of senders coordination (Table 7.1)	
Fault prevention	Achieved by a MAC protocol that coordinates access to the medium so that transmissions do not overlap and are scheduled according to timing requirements.
Fault removal	Not considered.
Fault tolerance	It is covered with redundancy in the time dimension. The applicable mechanism uses time diversity and considers timing requirements.
Fault forecasting	Faults are injected into the system to study their impact and measure the performance of the system in handling them. However, the frequency and duration of faults in a real system have not been analyzed due to complexity reasons, as they are highly dependent on the specifics of the scenario, and also for time limitations.

7.4.3 Summary

This contribution concludes after outlining the technologies and mechanisms that the solution in this thesis relies on to fulfill the requirements. To summarize, the deployment of mixed wireless and wired networks is based on IEEE 802.11 and Ethernet, respectively. The characteristics of IEEE 802.11 and Ethernet enable the fulfillment of several requirements: large data size, high throughput, wide range and scalability. Both technologies were developed for joint use in IT, which favors their compatibility. Since both are implemented in a wide range of COTS devices, the costs are low and the technologies are proven, which could promote ease of maintenance. IEEE 802.11 and Ethernet already cover the physical and partially the data-link layer, which reduces

Table 7.9: Description of the fault handling by the proposed solution (2/3).

Fault 2: lost transmission due to path loss, shadowing, multipath fading, overlapping transmissions and/or interference (Table 7.2)	
Fault prevention	Achieved through a MAC protocol that coordinates access to the medium so that transmissions do not overlap and are scheduled according to their timing requirements. In addition, the use of frequency diversity in the channel with the cognitive radio approach increases the probability of a successful transmission.
Fault removal	Not considered
Fault tolerance	It is covered by redundancy in the time dimension. The applicable mechanism uses time diversity and takes into account the time requirements.
Fault forecasting	Faults are injected into the system to study their impact and measure the performance of the system in handling them. However, the frequency and duration of faults in a real system have not been analyzed due to complexity reasons, as they are highly dependent on the specifics of the scenario, and also for time limitations.

Table 7.10: Description of the fault handling by the proposed solution (3/3).

Fault 3: lost transmission due to broken wires (Table 7.3).	
Fault prevention	By using rugged wires.
Fault removal	Not considered.
Fault tolerance	It is covered with redundancy in the space dimension. The proposed mechanism utilizes a wireless backup network to provide an alternate data path for transmissions that would have otherwise been performed over the wired network.
Fault forecasting	Faults are injected into the system to study their impact and measure the performance of the system in handling them. However, the frequency and duration of faults in a real system have not been analyzed due to complexity reasons, as they are highly dependent on the specifics of the scenario, and also for time limitations.

the complexity of the proposed solution. The use of PTP supports the requirements for data time stamping and synchronization.

The solution proposes a TDMA-based wireless MAC protocol supported by the TTE traffic classes and various TSN mechanisms. A scheduling solution is proposed that takes into account various occurrence patterns and criticality of data while meeting delivery latency and jitter requirements, delivery deadline, and providing a scalable solution that works in multi-hop networks.

The TDMA-based wireless MAC protocol is complemented by fault prevention mechanisms, including cognitive radio, and fault tolerance mechanisms, including time redundancy. In addition, the wired segment benefits from a fault tolerance mechanism based on space redundancy that uses a wireless backup network. With these mechanisms, the requirements for reliability, availability, safety and determinism can be met.

Chapter 8

Contribution area 2: fault-prevention mechanisms for the wireless medium

8.1 Introduction

This contribution proposes a wireless communication solution that complements and extends wired networks and enables dependable real-time communications. The wired network relies on the dependable and real-time communication technologies TTE or TSN, both applicable to Ethernet. The proposed solution is compatible with both, but one of the two technologies must be selected before setting up the network. The wireless solution is composed of the physical and data-link layers defined by IEEE 802.11, and thereby uses COTS hardware. The choice of Ethernet and IEEE 802.11 as wired and wireless technologies, respectively, enables high-throughput operation. The nodes are connected via a multi-hop switched Ethernet topology, where IEEE 802.11 can be used on the last hop.

Five MAC mechanisms in the wireless segment are proposed in this contribution area to offer a comparable level of reliability and real-time guarantees as in the wired segment. The mechanisms overcome the lack of timing guarantees of the baseline MAC in IEEE 802.11, i.e., DCF, and the potential problems of the wireless channel, such as path loss, shadowing, multipath fading and interference. The proposed MAC mechanisms follow a TDMA scheme, where each time slot refers to a specific point in time at which a transmission can take place. Time slots can also be reserved for procedures that are required for the MAC protocol to function, e.g., channel sensing as part of a cognitive radio protocol. The solution includes a scheduler which is responsible for appropriate time slot allocation, taking into account data exchange requirements, protocol-related constraints, and a multi-hop topology that enables the transmission of data over wired and wireless segments. The support for data transmissions with different timing, reliability and criticality requirements is accomplished thanks to their different treatment at the MAC layer. The different treatment results from the adoption of the TTE traffic classes TT, RC and BE, or the TSN mechanisms ST, CBS and legacy Ethernet traffic.

A guarantee for the access to the medium is necessary but not sufficient to enable reliable data transmissions, as path loss, shadowing, multipath fading and interference can still have a negative effect. To deal with these problems, this contribution area proposes an additional mechanism based on cognitive radio to increase reliability and reduce the likelihood of being affected by such circumstances.

The proposed MAC mechanisms are summarized in Table 8.1. The faults identified in the problem formulation in Section 7.2.2 which are addressed by the proposed MAC mechanisms are summarized in Table 8.2. Finally, the research questions described in Section 1.3 that are addressed by the proposed MAC mechanisms are included in Table 8.3.

Table 8.1: Summary of the MAC mechanisms proposed in Contribution area 2.

Group	**Label**	**Description**
Fault prevention in the critical-traffic handling (described and analyzed in Section 8.4)	`MAC_CH`	Scheduled time slots for critical traffic
Fault prevention in the best-effort handling (described and analyzed in Section 8.5)	`MAC_BH_1`	Pre-scheduled BE time slots with a round-robin assignment
	`MAC_BH_2a`	Contention-based BE time slots with round-robin assignment for a prioritized node. (a) No CW.
	`MAC_BH_2b`	Contention-based BE time slots with round-robin assignment for a prioritized node. (b) With CW.
	`MAC_BH_3a`	Contention-based BE phases. (a) Backoff time counter kept between the phases.
	`MAC_BH_3b`	Contention-based BE phases. (b) Backoff time counter reset between the phases.
Fault prevention using cognitive radio (described and analyzed in Section 8.6)	`MAC_CR`	Selects the frequency that attempts to maximize the probability of success of a transmission.

Note that although the mechanisms proposed in contribution areas 2 and 3 have been evaluated individually in most cases, they can also be combined if and when required by the requirements of the applicable scenarios.

Table 8.2: Faults (described in Section 7.2.2) addressed by the MAC mechanisms proposed in Contribution area 2. An "x" in the table means that the fault is handled by the MAC mechanism.

MAC mechanism	MAC_CH	MAC_BH_*	MAC_CR
Fault 1: delayed transmission due to lack of senders coordination (Table 7.1)	x	x	
Fault 2: lost transmission due to path loss, shadowing, multipath fading, overlapping transmissions and/or interference (Table 7.2)	x	x	x

Table 8.3: Research questions (described in Section 1.3) addressed by the MAC mechanisms proposed in Contribution area 2. An "x" in the table means that the research question is addressed by the MAC mechanism.

MAC mechanism	MAC_CH	MAC_BH_*	MAC_CR
RQ2.1: What fault-prevention mechanisms can be used to support reliable and real-time communications for the wireless medium?	x	x	x

8.2 Related work

8.2.1 Critical-traffic handling mechanisms

The handling of critical traffic having dependability and real-time requirements over technologies such as IEEE 802.11 or IEEE 802.15.4 has been subject of extensive research. Most works adapt the MAC layer to provide bounded latencies and avoid collisions between the nodes belonging to the network.

The authors in [75] present a token-passing approach. To protect traffic from external CSMA/CA transmissions, the highest priority level EDCA is used, while external nodes are expected to use the regular priority level. However, as mentioned in the baseline MAC protocols section (Section 2.2.2), token-passing approaches have significant drawbacks, including the token circulation overhead and the need to perform a token recovery procedure every time the token is lost.

The work in [76] presents a polling-based MAC with a dynamic adaptation mechanism that adjusts the number of polling messages sent to each node. The adaptation mechanism checks the node's history of previous data transmissions and counts the number of empty messages it has sent, with a larger number of them indicating that the node does not have much data to transmit. The focus is on reducing the number of unused polling rounds resulting from inflexible round-robin slot assignments. Unfortunately, the polling messages cause a noticeable overhead, as already stated in the baseline MAC protocols section (Section 2.2.2).

The authors of [77] focus on improving the admission and scheduling algorithms of HCCA and prove that better performance can be achieved if the admission and scheduling algorithms are designed to match the specifics of the actual use case. The slot assignment for HCCA is performed following the EDF scheduling policy, with schedules that are calculated in advance so that they are immediately available in case of dynamic changes in the network. Unfortunately, the EDF-based policy is not able to create configurations for multi-hop topologies by default.

The authors in [78] also propose a scheduling policy based on EDF, but allowing dynamic changes after a consensus process where the nodes exchange state information, including their requests for data exchange.

The work of [79] presents a MAC protocol based on maintaining a separate control channel in which nodes exchange information about their data exchange requirements. Based on these requirements, all nodes create an EDF schedule for the data exchange.

Similarly, [80] uses an EDF scheduling policy to select the next transmission. Timing guarantees can be given as long as the EDF schedulability test is passed.

The work in [81] presents the so-called IsoMAC approach, which is based on TDMA. In IsoMAC, the time between two beacon frames is divided into a scheduled phase and a contention phase. The scheduled phase is in turn divided into time slots that are allocated based on the requests sent by the end system nodes to a coordinator. The schedule defines a downlink phase for transmissions between the coordinator and the data destination nodes and an uplink phase in the opposite direction. The schedule is transmitted via the beacon frame. Prioritization over DCF traffic is achieved by separating the frames with SIFS instead of DIFS. As with IEEE 802.11, feedback on transmission success is based on ACK frames. However, these are not sent immediately after the data transmission, but the acknowledgment of the downlink frames is postponed until the uplink phase. The uplink frames are not acknowledged, but the coordinator uses the information contained in the schedule to detect failed frames. This way of handling the ACKs serves to reduce the protocol overhead. IsoMAC outperforms the IEEE 802.11 standard in terms of delivery latency jitter. Unfortunately, the dynamic generation of a schedule is based on end-system requests provided via contention, which means that the chance to utilize time slots is not guaranteed. In addition, the protocol imposes fixed phases for uplink and downlink transmissions, which can impair the schedulability of the network due to their lack of flexibility. Such an impairment is particularly relevant for scenarios with hybrid wired/wireless multi-hop topologies.

Similarly, the works in [82] and [83] propose a slot allocation in cycles that repeats the sequence of downlink, uplink and BE transmissions. Unfortunately, such a structure has a negative impact on the schedulability of multi-hop deployments.

The authors in [84] present a TDMA protocol with an online allocation of time slots based on the requirements specified by the nodes when joining the network. The schedule is conveyed using the beacon frame, leaving the specific scheduling policy to the user. The real-time guarantees therefore depend on the chosen scheduling policy. Feedback on the success of the transmission is sent within the time slot. Retransmissions can take place within the time slot or in another time slot if one is free. Latency is reduced compared to standard DCF and the packet loss ratio is improved for real-time traffic in the scenario studied, an office where significant interference can occur.

Another TDMA protocol is presented by [85]. In this work, the time dedicated to high-priority traffic is divided into time slots. Within the time slots, the medium is accessed using a higher priority class of EDCA than that used for legacy traffic. A feature of this protocol is that it allows direct transmissions between the end systems, avoiding the duplication of transmission that occurs when communicating via an AP. Unfortunately, the use of EDCA does not guarantee predictable medium access.

In [86], the authors propose a MAC protocol that distinguishes between real-time and non-real-time transmissions. For real-time transmissions, prioritized channel access is provided so that the backoff time is zero in the event of a collision. However, waiving the backoff time is no guarantee that the real-time transmission will take place. To sort the transmissions of real-time data, a fixed-priority scheduler with the rate-monotonic priority assignment algorithm is used. In addition, the time slots can contain multiple prioritized transmissions and retransmissions. If some of them are not needed, non-real-time traffic can be transmitted instead. Unfortunately, the rate-monotonic priority assignment algorithm can only handle transmissions over a single collision domain and is not easily translated to multi-hop topologies.

The TDMA protocol presented in [87] creates a transmission schedule that adapts to the number of users that can exchange data in a user-defined time period. The slots take into account the transmission time required by each of the users. Although the approach aims to avoid collisions, the round-robin channel access only works for a single collision domain and results quite limited to function in multi-hop topologies.

In [88], the authors show some preliminary simulation results on how safety-relevant use cases in industrial wireless environments can be covered by using TSN. Specifically, they use IEEE 802.11 as the wireless technology and combine it with the TSN mechanisms of ST and clock synchronization in an attempt to achieve predictable data delivery delays. The authors emphasize that special care must be taken to avoid sudden delays due to the IEEE 802.11 MAC and mechanisms such as retransmissions, random backoff times and beacons.

The work in [89] presents the so-called RRMAC, a TDMA protocol for WSN that executes a superframe structure between beacon frames with contention-free and contention-based access periods. Similar to other MAC protocols in WSN, there is an additional inactive period in which the nodes do not communicate to save energy. During the contention phase, the contention-free time slots are assigned to each node. This means that a chance to get access to the contention-free slots is not guaranteed, as contention can prevent a node from gaining access to a time slot. In WSN, it is common to apply strategies for more efficient data collection. In the specific case of RRMAC, communication between nodes follows a tree structure in which leaf nodes send their data to the nodes connected to them and aggregate the data in a process that continues until all branches converge in the root of the tree, which is the final destination of all data.

The so-called source-aware scheduling algorithm (SAS-TDMA) [90] proposes a TDMA protocol for WSN that adapts to network conditions and attempts to provide a fast response to changes in data routes. The algorithm uses information from the physical, link and network layers to create the schedule that allocates the time slots. The schedule may be updated if a change in the routes is detected, with an algorithm that attempts to adapt to these changes quickly and with a small configuration overhead. As usual in WSN use cases, the data is retrieved from the sensors and collected in a sink node, leaving out other node interactions.

The work in [91] proposes a MAC protocol for WSN in which periodic, non-critical traffic gets assigned slots, but these can be reallocated for critical aperiodic data transmissions. If two or more data transmissions have the same criticality level, the one closest to its deadline is given access. The mechanism is limited to star topologies where all nodes are in the same collision domain, so hybrid wired/wireless multi-hop topologies are excluded.

In the work by [92], the authors present another example of a TDMA protocol for WSAN with contention and contention-free phases, and a beacon frame to convey the schedule. In this protocol, retransmissions are performed in the contention-free phase to increase reliability. The protocol limits its topology options to a single star.

The authors in [93] propose a MAC protocol for WSN over mesh topologies. The work introduces a TDMA schedule that is placed on top of the multi-hop topology. The schedule is dynamically created based on the retrieval of the topology description by a master node. Once the schedule is created, it is sent to the nodes for them to follow it. Updates to the schedule due to topology changes are possible. The schedule divides the available time into a slot for control transmissions and slots for data. The control slot is used by the protocol, e.g., to retrieve the topology updates, exchange the schedule with the nodes or for synchronization purposes. The protocol supports time redundancy and space redundancy, the latter by transmitting data via different routes. Unfortunately, the scheduling solution for the data transfers is not specified, so the fulfillment of real-time guarantees remains open.

The so-called dual-mode real-time MAC protocol [94] presents an approach for WSN that avoids collisions and includes relaying, but requires nodes to know their position in order to decide when to relay.

The work in [95] proposes a mechanism for accessing the medium that is similar to the CAN arbitration phase. Before transmitting data, each node uses its pre-assigned arbitration frequency to deterministically decide which node has the right to transmit. Such an arbitration phase enables a decentralized mechanism to decide which node accesses the medium and does not require any precomputed schedule. However, the arbitration process can be time consuming and might neglect transmissions from less prioritized nodes.

The work in [96] proposes a mechanism to gain access to the medium in environments where CSMA/CA is applied by sending an interfering signal to clear the channel before a transmission.

In [97], the authors propose a TDMA-based MAC mechanism for WSAN. The MAC divides the available time into ten time slots, each of which can host a single data transmission. Each of these slots is then subdivided into ten subslots. Access to the slot is based on a process in which a beacon is first sent to announce that the node has something to transmit. Data transmission then takes place in the next subslot. Access to the slots is based on priorities, with the nodes having the chance to send the beacon in a sequence determined by their priority. If the beacon is not sent, the node with the next priority level is given the chance to send the beacon in the next subslot. The process is repeated for the different priority levels defined in the protocol. The scheduling mechanism focuses on providing a better medium access time to the highest priorities, but to do so it imposes a slot structure that does not take into account other aspects such as the timing requirements of data exchanges or the integration into a multi-hop network topology.

8.2.2 Critical-traffic handling implementations over hardware

Several academic works have proposed real-time wireless solutions based on IEEE 802.11 and implemented them in hardware. Most of them suggest TDMA schemes to organize access to the medium. The access is often laid out in cycles, with some time slots precisely allocated and others accessible via CSMA/CA. The actual schedule of slots differs from using EDF [98], round-robin [99] or time-triggered [100][84][101][102]. The work in [103] provides a framework in which different scheduling policies can be applied and relies on the so-called one-shot mechanisms of the hardware.

The synchronization of the nodes to follow the TDMA scheme varies between the proposals. In [98], a coordinator node uses a dedicated request transmission to trigger data transmissions, resulting in some overhead. In other implementations, synchronization with a coordinator node is done via the IEEE 802.11 beacon frame [84][101] or via special synchronization messages [99][100][102]. If the data transmissions are not explicitly triggered by a coordinator as in [98] or the rules for maintaining a schedule cannot be derived by the nodes themselves, a schedule is distributed to the nodes. In the reviewed works, the schedule can either be included as part of the beacon or sent in a special frame.

Several of the analyzed works present implementations that use modified versions of the `ath9k` driver. Their reliability and jitter performance numbers are quite promising when compared to standard DCF, but strongly depend on the interference scenario. The work in [98] shows that results of over 99% of frames delivered on time can be achieved when scheduled retransmissions are included. In [99], improvements in throughput and timeliness are achieved compared to similar use cases in factory automation. The proposal of [84], called RT-WiFi, brings significant improvements in terms of frame delivery latency compared to DCF after retransmissions are removed, but at the cost of losing between 7% and 10% of frames in office interference scenarios. An implementation of RT-WiFi that uses an SDR approach instead of COTS hardware is presented in [104]. The SDR option provides a hardware alternative in cases where there is no open source driver to control the hardware or parts of the required hardware functionality are not openly available. The authors in [103] utilize the built-in one-shot mechanism of Atheros chipsets to deliver 99% of frames with low jitter, similar to the numbers in [101], with 95% of frames delivered on time and µs range jitter in scenarios with moderate interference. In contrast, the work in [102] controls transmissions without specialized hardware support, causing a larger overhead.

The scheduling of the above protocols is intended for single-hop transmissions. The work in [105] takes a distinct multi-hop approach and achieves low packet loss ratios of about 1%, but its adaptation over the Atheros hardware uses only one transmission queue and does not allow different treatment of data based on its criticality. An even wider range is pursued in [106], which extends coverage to the WAN scope using a multi-hop protocol. The timing for the transmissions is managed by providing the data to the hardware immediately before sending. Their results show that µs granularity for the case of transmitting clock synchronization data.

Based on the introduction of TSN in wired deployments, development towards wireless solutions based on TSN has emerged in recent years. The contribution by [83] leaves out COTS equipment and opts for a new field-programmable gate array (FPGA)-based hardware design with the proposal of a new physical layer. The customized hardware exhibits promising results in terms of delivery latency, but is not yet compliant with TSN or IEEE 802.11. This wireless implementation is deployed together with a wired TSN network in [107] to demonstrate the feasibility of an integrated network and show that low latencies can be achieved even when exchanging data between the wired and wireless segments. In [108], the behavior of ST is emulated and some preliminary results are provided, focusing on showing the performance degradation in terms of delays compared to using ST in the wired network and the disadvantages caused by the lack of more precise control of the hardware. Finally, in [109], the authors present an implementation of ST over IEEE 802.11. Their results are mainly concerned with describing the impact of enabling ST on the behavior of a particular use case involving robots, lacking a comprehensive analysis of performance in terms of reliability and delays.

The potential lack of timing guarantees when the operating system takes part of the responsibility for communication is addressed in [110], which opts for the use of Xenomai, a Linux-based real-time operating system. The authors in [111] prove that TDMA over COTS hardware and a non-real-time capable Linux kernel are able to achieve a low data transmission jitter of about $10\,\mu$s and and data transmission periods of $256\,\mu$s with the support of the Linux high-resolution timer.

8.2.3 Clock synchronization over the wireless medium

Some types of TDMA systems are based on schedules for accessing the medium, which require synchronization of the nodes. At the same time, synchronized nodes also enable timestamping of events. The work in [112] provides an overview of clock synchronization options over IEEE 802.11, which include the built-in IEEE 802.11 TSF, PTP and Network Time Protocol (NTP). The article emphasizes the importance of node timestamping capabilities to achieve different levels of precision and points out that hardware timestamps are always preferred when nanosecond precision is needed. However, [113] shows that even in the case of software timestamps, several adaptations can be made to PTP to improve its performance in IEEE 802.11 networks. Some of these adaptations include the transmission of timing information using beacon frames or mechanisms to compensate for path delays. The work of [114] demonstrates how clock synchronization based on IEEE 802.1AS can be adopted with sufficient precision in hybrid wired and wireless networks based on Ethernet and IEEE 802.11, respectively.

8.2.4 Cognitive radio

Several cognitive radio based solutions for real-time communication systems have been presented to cope with interference in ISM bands. The work in [115] presents the so-called CH-MAC, which uses a common control channel (CCC) to share control information and make decisions about which channel to use for transmissions. A similar proposal was made in [116], but in this case the nodes have two antennas. One of the antennas operates in a dedicated CCC and the other antenna is used for data transmissions and can be reconfigured dynamically to transmit in the most adequate frequency. However, both MAC protocols are based on a CSMA method, so no bounded channel access time is provided. In [79] a TDMA approach for the CCC is presented, where the control data is transmitted to an AP, which decides on the channel to be used. However, all these MAC schemes depend on a CCC, which may be of poor quality due to interference, making it difficult to exchange control information. Therefore, some other designs have been proposed which do not depend on a CCC.

In [117], a specific period of time is reserved for the exchange of coordination data between the devices, so that the use of a CCC is not required. The timeline is divided into four different segments: sense, control, feedback and data. All devices receive information about the environment during the sensing time. Then this information is sent to the AP, which analyzes it. Afterwards the AP sends the decision of which channel to use for transmissions in the feedback period, and finally the devices access the medium to transmit data using a TDMA method.

A rendezvous process may be used so that all devices tune to the same channel. In the rendezvous process, the devices search for each other in the available channels, i.e., each device tries every channel until it encounters another device. In [118], a rendezvous sequence is proposed to ensure that two devices meet in at least one channel in a given time interval. Therefore, if a channel is available between the sender and the receiver, this MAC ensures correct reception.

In [119], some of these cognitive radio proposals were evaluated and compared with non cognitive radio based MAC schemes in industrial environments, proving that cognitive radio can be used to cope with interference in these environments.

Unfortunately, the described cognitive radio based MAC protocols have some limitations as they do not consider scheduling for operating in the context of a multi-hop wired-wireless networks with support for traffic with diverse timing requirements.

8.3 System model

8.3.1 Topology

The topology of TTE and TSN is based on switched Ethernet, with networks comprised of end systems as the source or destination of data, and switches as intermediate nodes. Each end system is connected to a switch via a full-duplex link, establishing a star topology. Switches are not limited to connecting only end systems, but can also be connected to each other, turning the network topology into a star where each of the star leaf nodes can be the central point of another star topology. The protocols that support Ethernet, e.g., for routing, restrict the logical topologies to avoid cycles that could cause problems such as a broadcast storm.

On the wireless side, star topologies are as well adopted based on the IEEE 802.11 infrastructure topology. The similarity of the topology is advantageous for the integration of the wired and wireless segments, as it does not bring more complexity for the scheduling of data traffic. The wireless end systems communicate via the AP, which acts as the central node in the infrastructure topology. The AP is also responsible for the interface between the wireless and wired segments. Figure 8.1 shows an example of the topology, where wired full-duplex connections define two collision domains on each physical link, while each AP and its associated end devices form a single collision domain in the wireless segment. All wireless end systems must be installed in such a way that they are within range of the corresponding AP. Areas that are covered by different APs but overlap must avoid mutual interference by either using different frequencies or coordinating transmissions, e.g., through a MAC that takes the coordination responsibility.

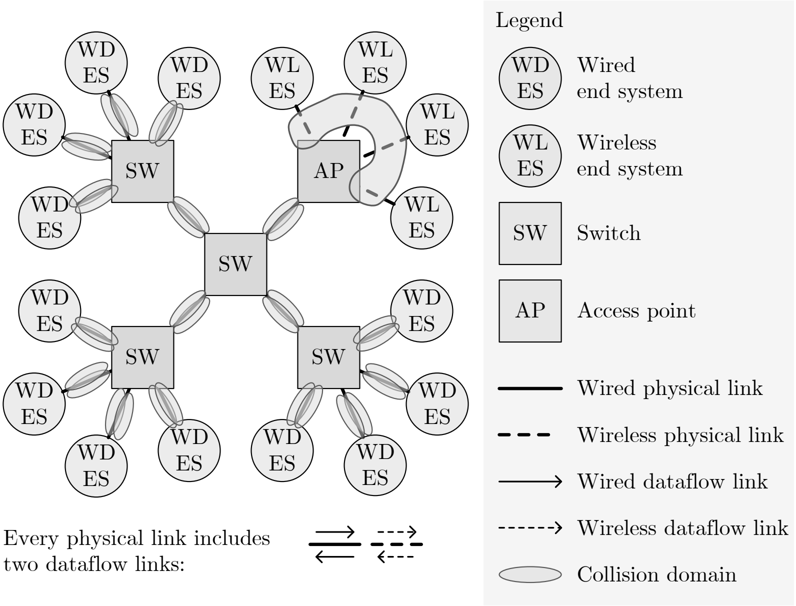


Figure 8.1: Topology model for the mechanisms in Contribution area 2.

The physical topology of this multi-hop network is given by the undirected graph $G(V, E)$, where V stands for the network nodes and E represents the communication links between the nodes. In the network under consideration, the physical links are bidirectional and are referred to as dataflow links. The set of dataflow links is labeled as L, so that

$$\forall [v_i, v_j] \in V : (v_i, v_j) \in E \Rightarrow [v_i, v_j], [v_j, v_i] \in L, \tag{8.1}$$

where (v_i, v_j) denotes an undirected edge, i.e., a physical link, and $[v_i, v_j]$ denotes a directed edge, i.e., dataflow link.

8.3.2 Traffic

Support for applications with diverse time and safety requirements in TTE is ensured by three different traffic classes, as already described in the TTE section (Section 5.2.1). The TT class, which is used for data exchanges that require guaranteed latency with low jitter, is more suitable for periodic data exchanges. The guaranteed latency is possible if the data generation takes place periodically and is synchronized with the data handling. This ensures that both generation and handling do not deviate from each other. If such a drift occurs, an offset between generation and handling accumulates over time, which means that the delivery jitter cannot be kept low. Sporadic data exchanges served by the TT class do not benefit from low jitter for a similar reason as periodic unsynchronized exchanges: The time-triggered slot scheduling is fixed, while the sporadic data generation instants are characterized by their variability. In the TT class, the maximum difference between data generation and its handling is given by the handling or generation period, which should in principle be the same. The worst case when waiting for a handling opportunity occurs when the generating application misses the current data handling slot right when the slot is over and has to wait for the next one. Nevertheless, the RC traffic class offers more relaxed timing guarantees than the TT traffic class, with a latency that remains bounded, but no guarantee of low jitter. The RC handling mechanism is suitable for data exchanges generated periodically or sporadically. Since TT and RC traffic classes offer timing guarantees, they are the best candidates for ensuring critical data exchanges.

The BE traffic class is expected to be used when time guarantees are not required and data exchanges are not critical. Contrary to what is often assumed, BE data exchanges do not have to be limited to an unbounded data generation pattern, but can also be generated according to periodic or sporadic patterns. The non-consideration of BE data exchanges in the modeling the system is more related to their lack of criticality, as some of the BE exchanges could very well be generated periodically, which would provide the necessary information for the configuration of the system to handle them with guarantees, if required. In TTE, BE uses the time remaining after allocating the communication resources for the transmissions of TT and RC traffic classes.

In the proposed mechanisms, a scheduler plans when TT and RC transmissions shall take place. The remaining time can then be used by BE traffic. Various mechanisms for using the time left for BE traffic over the wireless medium are presented later. Table 8.4 summarizes the characteristics of the TTE traffic classes.

Table 8.4: TTE traffic classes

Traffic class	**Handling occurrence pattern**	**Possible generation occurrence pattern**	**Criticality**	**Timing guarantees**
Time-triggered (TT)	Periodic	Periodic	Critical	Guaranteed latency
Rate-constrained (RC)	Sporadic	Periodic, sporadic	Critical	Bounded latency
Best-effort (BE)	Unbounded	Periodic, sporadic, unbounded	Non-critical	None

In the case of TSN, there are no traffic classes that are explicitly defined by the provided standards. Rather, TSN defines a set of data handling mechanisms. The TSN mechanisms theoretically allow any combination, but in practice some combinations of mechanisms work better than others for each use case. Therefore, a different combination of mechanisms results in different handling of data transfers, each targeting a different set of requirements. Since TSN is usually applied over Ethernet, a straightforward traffic class would be the one supported by standard Ethernet and based on plain CSMA/CD medium access. Such a traffic class is referred to as BE in the proposed mechanisms. Just like BE traffic class in TTE, the BE traffic class in TSN takes the remaining time left after the transmissions of traffic classes with a higher criticality.

Adopting the scheduled traffic mechanism from IEEE 802.1Qbv potentially enables data transmissions with guaranteed latency and low jitter suitable for periodic critical data. In the proposed mechanisms such a traffic class is referred to as ST. ST is similar to the TT traffic class of TTE with some differences due to their different implementation, which results in worse delivery jitter for ST than for TT traffic. However, for the sake of simplicity, the proposed mechanisms assume similar timing guarantees. As with TT, the periodic data generation and the handling based on ST must be synchronized to ensure the lowest possible jitter.

The existence of BE and ST classes is the bare minimum in order to treat data transmissions with different criticality in a differentiated manner. A third and optional traffic class based on the CBS mechanism of IEEE 802.1Qav could be offered, provided it is available in hardware. The CBS class of TSN is similar to the RC class of TTE. CBS is capable of exchanging critical data that is generated either periodically or sporadically, but in contrast to TT/ST, the delivery jitter requirement is relaxed and only bounded latency is offered. The data served by CBS could also be handled by ST, as the ST mechanism can reserve the required bandwidth. However, ST and CBS handle the bandwidth reservation differently. The former reserves time slots for ST, that might be reused by other transmissions if the intended ST is not sent. In contrast, CBS only uses up the credit when data is transmitted, but when CBS data is not sent, other traffic can use the bandwidth.

Additional TSN mechanisms can be selected to define new classes. For example, a new and more reliable traffic class for the exchange of critical data could use the mechanism of “Frame Replication and Elimination for Reliability” (IEEE 802.1CB). However, it is considered that the three classes described are sufficient to meet the requirements and provide a similar offering to TTE in the proposed mechanisms. Table 8.5 summarizes the characteristics of the TSN traffic classes supported by the proposed mechanisms.

Table 8.5: A proposal for TSN traffic classes

Traffic class	**Handling mechanism**	**Handling occurrence pattern**	**Possible generation occurrence pattern**	**Critical.**	**Timing guarantees**
Scheduled traffic (ST)	Scheduled traffic (IEEE 802.1Qbv)	Periodic	Periodic	Critical	Guaranteed latency
CBS	CBS (IEEE 802.1Qav)	Sporadic	Periodic, sporadic	Critical	Bounded latency
Best-effort (BE)	Legacy Ethernet	Unbounded	Periodic, sporadic, unbounded	Non-critical	None

The traffic model for the network takes into account the three traffic classes from TTE and their proposed equivalents for TSN. Their characterization follows the one previously given in the baseline characterization of events (Section 4.1.1) for periodic, sporadic and unbounded data transmissions. A TT/ST message m_i is therefore characterized by the tuple (C_i, D_i, T_i, Φ_i). A RC/CBS message m_i is characterized by the tuple (C_i, D_i, f_i) or (C_i, D_i, T_i^{min}). Finally, a BE message m_i is characterized by the tuple (C_i, D_i). C_i is often set to match the data delivery latency estimate. The data can be generated according to a pattern that differs from the handling pattern of the corresponding traffic class. If this is the case, the data transmission requirements must still be specified according to the characterization of the handling pattern of the selected traffic class. For example, for sporadically generated data handled by TT/ST, a time period T shall be specified, calculated to correspond to the generation frequency f or the minimum inter-arrival time T_i^{min}.

A message m_i is generally transmitted several times, e.g., after the period elapses in a periodic data exchange. Each of these transmissions is referred to as an instance $m_{i,j}$, $j = 0, 1, 2, ...$ Then a message instance that needs to be transmitted from a sender to a receiver is decomposed into a set of message instances $m_{i,j}^{[v_j,v_k]}$, one for each of the dataflow links $[v_j, v_k]$ over which it needs to be transmitted to reach the destination. The network schedule includes the instances of all TT/ST and RC/CBS messages $M = m_1..m_n$. These messages are considered critical. In addition, $O_{i,j}^{[v_j,v_k]}$ represents the time at which a message instance is scheduled to be transmitted over a dataflow link $[v_j, v_k]$, formulated as an offset from the time at which the schedule begins. The task of calculating $O_{i,j}^{[v_j,v_k]}$ is performed by the scheduler.

8.4 Fault prevention in the critical-traffic handling (MAC_CH)

8.4.1 Mechanism description

Supporting the guaranteed latencies of TT/ST and the bounded latencies of RC/CBS traffic requires a coordination mechanism at the MAC layer. The coordination mechanism must prevent the transmissions of these traffic classes from overlapping, otherwise reliability will be compromised and timing requirements may not be met (as initially indicated in Table 7.8 and Table 7.9 in the fault-handling proposals). Therefore, a TDMA-based MAC labeled MAC_CH is proposed [1][2][3][7], where the available transmission time over the wireless medium is divided into time slots.

The time slots are allocated to the different traffic classes based on their timing requirements using a scheduler. In the proposed mechanism, the TT/ST is scheduled offline. In the worst case, the RC/CBS flows are periodic, so they can also be modeled as TT/ST flows and added to the offline scheduler. Thus, the access to the wireless medium for TT/ST and RC/CBS traffic is bounded and contention-free, i.e., deterministic. In this way, collisions due to uncoordinated medium access are avoided. The BE data takes up the remaining time after the allocation of TT/ST and RC/CBS traffic. The generated schedules, which take into account the handling of all traffic classes, are distributed to the participant nodes before the network is put into operation. Further details on scheduling can be found later in Section 8.4.2. Clock synchronization, which provides the network nodes with a common notion of time and enables time-triggered schedules to be adhered to, is ensured by PTP. The data exchanges required by the PTP protocol are handled as BE data.

Algorithm 8.1 contains the pseudocode that describes the handling of critical data traffic. The procedures from the pseudocode are clarified in Appendix A. The pseudocode shows how the slots are assigned to the data exchanges requested by the applications (lines 3-4).

Algorithm 8.1 Fault prevention in the critical-traffic handling: Pseudocode describing the MAC mechanism `MAC_CH`.

```
1: procedure MAC_CH
2:     while true do
3:         wait_for_assigned_slot()
4:         if is_current_slot_critical() then
5:             if not is_critical_queue_empty() then
6:                 $m$ ←dequeue_critical()
7:                 send_asap($m$)
```

The time T_{slot} required for the slots must take into account the data to be transmitted T_{data} plus a guard interval chosen to correspond to the slot time T_{ST} used to avoid collisions due to propagation delays. Based on the individual generation periods of TT/ST and RC/CBS traffic, a hyper period is defined as the least common denominator of the data generation periods. The schedule is to be repeated continuously during the runtime of the communication system in cycles of the duration of a hyper period. A detailed description of the scheduler for handling critical traffic can be found in the upcoming section.

The use of DCF mechanisms such as NAV or RTS/CTS is not required in the proposal, but could be introduced if needed by considering longer slots that can accommodate the additional overhead. ARQ is also not required, but when implementing the proposal, mechanisms such as ACKs might not be easily disabled. Table 8.6 summarizes the DCF and ARQ mechanisms that apply to the proposal for handling critical traffic.

8.4.2 Scheduling

Scheduling goal

The described fault-prevention mechanism for the wireless medium, which handle different types of traffic, require scheduling the utilization of the limited communication resources. The resources are shared by multiple nodes sending and receiving data such that:

- Faults are prevented when handling critical traffic by means of the `MAC_CH` mechanism. TT/ST and RC/CBS traffic is transmitted over a hybrid wired and wireless multi-hop network, taking into account their data size, occurrence pattern and delivery deadline requirements.

Table 8.6: Fault prevention in the critical-traffic handling: summary of DCF and ARQ mechanisms applicable to the proposal.

MAC mechanism	`MAC_CH`
Traffic type	**Critical**
$\boldsymbol{T_{DIFS}/T_{AIFS}}$ **sensing required?**	No
Minimum required $\boldsymbol{T_{slot}}$	$T_{ST} + T_{data}$
CW required?	No
Backoff required?	No
NAV required?	No
RTS/CTS required?	No
ACK required?	No
ARQ required?	No

Specifically, the scheduler is responsible for determining $O_{i,j}^{[v_k,v_l]}$, which specifies the time at which a message instance $m_{i,j}$ must be transmitted over a dataflow link $[v_k, v_l]$, formulated as the offset from the time at which the schedule begins. If the scheduler is used for TSN, it must also configure in which of the eight available queues the data shall be stored before transmission.

For very simple topologies and data exchange scenarios, creating a schedule that meets reliability and real-time requirements might be simple, even to the extent that it can be created by hand or by easy-to-use configuration algorithms, e.g., fixed-priority scheduling using rate monotonic priority assignment. However, in the scenarios considered in this contribution, with real-time data transmitted over multi-hop topologies, finding schedules becomes exceptionally challenging. Inspired by the proposals for TTE [11] and for TSN [57], the scheduling problem is formulated with a mathematical model of the network topology and traffic based on FOL constraints. The FOL constraints can also be used to model the proposed fault-prevention mechanisms. After formulating the scheduling problem, the schedule is generated by solving the mathematical model using SMT solvers [120]. In some cases, the scheduler is not able to obtain a schedule, which might require to reconsider the input data.

The output of the scheduler, i.e., the schedule, contains the configuration required for the above mechanisms for each of the nodes in the network. The schedule is generated by a computer and then made available to the nodes. Since the creation of the schedule is very complex and could take from minutes to hours, the schedules are created and distributed offline before the network is put into operation.

Background work: the TTE and TSN scheduling solution

The previously cited articles [11] and [57] describe scheduling FOL constraints for TTE and TSN, respectively. Some constraints are equivalent between both technologies, as they refer to aspects that are common to both, while others differ between TTE and TSN. The constraints are used to model the scheduling of TT/ST and can also reserve bandwidth for RC/CBS if made periodic. Below is an explanation of what each constraint models. The mathematical expressions used to formulate the constraints can be found in the corresponding articles.

The following constraints are present in both TTE and TSN:

- **Frame constraints**. Model that the offset to be calculated must be equal to or greater than 0, and that the data transmission has sufficient time to be completed during the period. To define these constraints the specification of the data exchanges is required.

- **Contention-free constraints**. Model the exclusive use of dataflow links by one transmission at a time. Consequently, two or more transmissions are not scheduled simultaneously on the same dataflow link. To define these constraints the network topology and the specification of the data exchanges are required.

- **Path-dependent or flow-transmission constraints**. Model that data is available on a node before it is transferred to another node. Without these constraints, the scheduler assumes that the data is available on all nodes from instant 0. Thus, these constraints guarantee that all frame instances that are sent via the dataflow links between a sender and a set of receivers are scheduled in a chronological order. To define these constraints the network topology and the specification of the data exchanges are required.

- **End-to-end transmission constraints**. Model the deadline requirement for data exchanges by specifying the maximum time that could be needed to transmit data from a sender to a receiver. To define these constraints the network topology and the specification of the data exchanges are required.

- **Bounded-switch memory constraints**. Model the data memory limitations on the intermediate nodes in the network. These limitations apply to data received and stored in these nodes while waiting to be transmitted. Without these constraints, the data could be stored in the switches without any limits, which is unrealistic. To define these constraints the memory capacity of the buffers on the network nodes, the network topology and the specification of the data exchanges are required.

The formulation of the constraints based on the FOL makes it possible to extend the defined set with additional constraints. For example, other constraints could also be defined to model aspects such as precedence so that the transmissions follow a customized order. Another example is the formulation of constraints to reserve bandwidth for clock synchronization, although synchronization protocols such as PTP can operate at the level of guarantees provided by BE.

The constraints that are different between TTE and TSN are due to the differences between the two technologies in the handling of data frames. These differences do not compromise the support for traffic classes but result in different performance. For example, a higher jitter can be expected with TSN than with TTE, as TTE has a finer control over the frame dispatching. Specifically, TTE uses a buffer to store the frames before selecting them for transmission, while TSN uses a set of eight FIFO queues. The former allows the selection of the next frame from the buffer that matches the VL identifier. In contrast, the FIFO queues in TSN only allow the selection of the first frame in the selected queue. Since there are often more streams than queues, different streams may have to share the same queue. As a result, the order in which the TSN FIFO queue is filled is of utmost importance, while the TTE buffer does not impose such a constraint. For example, the TSN queues that are fed with frames from the end systems could be filled in a non-deterministic way. To avoid such behavior, the end systems could control the dispatching of frames to constrain them to specific time windows or a TSN switch could serve as an interface to a single end system and control the frames injected into the TSN network. However, the latter option is likely more costly.

The following constraints are specific to TSN:

- **Egress-interleaving constraints**. Streams coming from different sources and having the same output port and priority in the switch might interleave and be queued together. However, the way in which they interleave may vary slightly in reality, e.g., due to clock synchronization precision, resulting in different interleaving and therefore a different order at the exit of the queues. Nevertheless, the scheduler selects the queue for transmission with the expectation that the selected frame belongs to a specific stream, which is not guaranteed if the order of queuing is not deterministic. These constraints provide a solution to this problem by avoiding frame interleaving. Scheduling guarantees that streams are either enqueued into different queues or, if they are enqueued into the same queue, the order in which they enter the queue is always the same by controlling the time that the preceding switches in the path of the stream dispatch them. To define these constraints the network topology and the specification of the data exchanges are required.

- **Stream-isolation constraints**. Dispatching a frame from a queue is correct as long as the first frame is the one that the scheduler was intended to dispatch, i.e., it belongs to the appropriate stream. Unfortunately, it can still happen that, e.g., the application generating the data does not make it available to the network at the expected time and the frame in question is not available for dispatch. In this case, the next frame in the queue could take its place, breaking the assumptions of any subsequent GCL on the other switches in the path of a frame and causing non-deterministic data delivery. These constraints avoid the situation described by guaranteeing that a frame from one stream is not queued until all other frames from another stream expected within a time window have been dispatched and the queue is empty. Therefore, the schedule either dispatches a frame from the intended stream or no frame at all, but never a frame from a stream that is not the expected one. To define these constraints the network topology and the specification of the data exchanges are required.

- **Frame-isolation constraints**. The stream-isolation constraints are quite restrictive and limit the options for finding a feasible schedule. Therefore, these constraints are relaxed so that frames from different streams can be interleaved as long as they are not queued at the same time, i.e., if a frame from a stream is queued, the queue must be empty. This is in contrast to the stream-isolation constraints, which require that all frames of a stream must be first queued and then dispatched before another stream can gain access to the queue. To define these constraints the network topology and the specification of the data exchanges are required.

The number of queues in TSN is limited by the eight priorities specified in the IEEE 802.1Q standard. However, the set of queues is often partitioned to serve different purposes, i.e., traffic classes, which may result in only a subset of the eight queues being available for ST. For example, the eight queues can be split between the supported traffic classes (Figure 8.2): ST, CBS and BE. Therefore, the constraints must take into account that only a limited number of queues might be available for ST. The work in [57] shows that reducing the number of queues available for ST has a negative impact on the number of possible solutions that can be found for the scheduling problem, but that it may still be feasible to find a solution.

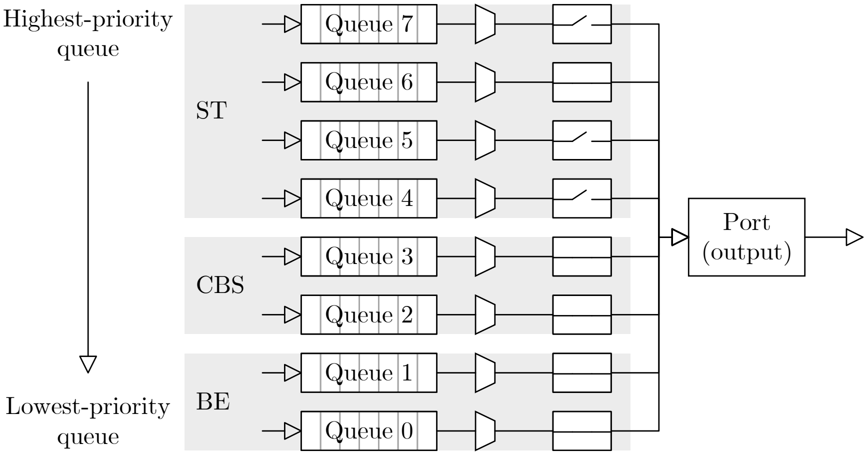


Figure 8.2: Fault prevention in the critical-traffic handling: Example of the assignment of TSN mechanisms to queues. In the example, the ST TG for queue 6 is open, but also the TGs for queues 0-3, which are used for CBS and BE traffic. In this way, the frames from the ST are selected first, if there are any. Otherwise, the CBS and BE frames in this order are selected for transmission.

If different queues are assigned to different traffic classes, it is possible to keep several TGs open at the same time without negatively affecting the traffic classes with higher criticality. The frames for transmission are selected on the basis of the TG with the highest priority among the open TGs that also have data to transmit. So if ST queues are assigned the highest priority, they will always be dispatched when the schedule opens their TGs. At the same time, the TGs of lower priority queues can be open and take the chance to transmit when there is no ST to transmit. Consequently, the frames of a lower priority can take over the slot so that no bandwidth is wasted. Following this reasoning and assigning the middle priority range to the CBS, CBS-ruled data will be transmitted in the remaining time left by the ST. Unfortunately, CBS may not get enough bandwidth, so either all critical data is mapped to ST and CBS is not used, or an analysis is performed to evaluate the blocking effect of ST on

CBS. The work in [121] investigates whether timing guarantees can be provided to CBS traffic after higher-priority ST is allocated. Finally, BE traffic can be assigned the lowest priority range so that it is only sent when ST and CBS traffic are not.

Integration into the TTE and TSN scheduling solution to support fault prevention in the critical-traffic handling

The constraints for TTE and TSN require an extension to take account of the particularities of the wireless medium. Specifically, it is necessary to redefine the contention-free constraints to model the broadcast nature of the wireless medium. Such constraints shall not allow concurrent transmissions on the dataflow links in each wireless collision domain, unlike the full-duplex links of Ethernet, where simultaneous transmissions are possible in the two directions of the link.

To introduce the new constraints, the formulation of the contention-free constraints from TTE and TSN is first reviewed, based on the definition from [11]. Given M representing the set of all critical messages m_i, T representing the set of all periods T_i, T_{slot} as the time required for a transmission plus overhead, and $LCM(T)$ which stands for the least common multiple of all T, the following equation models the contention-free constraints for TTE and TSN:

$$\begin{aligned}
&\forall [v_k, v_l] \in L, \forall m_i, m_j \in M, \\
&\forall a \in \left[0, 1, ..., \left(\frac{LCM(T)}{T_i} - 1\right)\right], \\
&\forall b \in \left[0, 1, ..., \left(\frac{LCM(T)}{T_j} - 1\right)\right] : \\
&((m_i \neq m_j) \land \exists m_i^{[v_k,v_l]} \land \exists m_j^{[v_k,v_l]}) \Rightarrow \\
&((a \times T_i) + O_i^{[v_k,v_l]} \geq \\
&(b \times T_j) + O_j^{[v_k,v_l]} + T_{slot}) \\
&\lor \\
&((b \times T_j) + O_j^{[v_k,v_l]} \geq \\
&(a \times T_i) + O_i^{[v_k,v_l]} + T_{slot}).
\end{aligned} \tag{8.2}$$

In principle, it is assumed that the message instances $m_{i,x}$ are always one period T_i apart, so that the offset $O_i^{[v_k,v_l]}$ only needs to be calculated once for all instances. For the rest of the message instances $m_{i,x}, x = 2..n$, an offset is expected, which is given by $O_{i,x}^{[v_k,v_l]} = (x-1) \times T_i + O_i^{[v_k,v_l]}$. The same principle applies to $m_{j,x}$ and $O_j^{[v_k,v_l]}$.

After reviewing the contention-free constraints for TTE and TSN, the following class of constraints is required to extend the support for critical traffic over the wireless medium:

- **Contention-free constraints for the wireless medium**. Model the exclusive use of wireless dataflow links by one transmission at a time. Thus, two or more transmissions are not scheduled simultaneously on the same dataflow

> link. In addition, these constraints take into account that no pair of wireless dataflow links can be scheduled to handle transmissions concurrently. These constraints could be generalized to scenarios with multiple wireless networks in non-overlapping frequencies. However, for simplicity, it is assumed that all wireless devices are in mutual range and consequently part of the same collision domain. To define these constraints, the network topology and the specification of the data exchanges are required.

To formalize the contention-free constraints for the wireless medium, L is redefined so that it is now the union of two sets: the wired dataflow links L_{wd} and the wireless dataflow links L_{wl}, such that $L = L_{wd} \cup L_{wl}$. T_{slot} can be set to different values depending on the specifics of the supported MAC mechanism or aspects such as the transmission rate of a given link, which may vary between wired and wireless media, for example. However, for the sake of simplicity, it is assumed that only a single value for T_{slot} is applicable. Consequently, the following equation describes the contention-free constraints for a network consisting of wireless links:

$$\begin{aligned}
&\forall [v_k, v_l], [v_q, v_r] \in L, \forall m_i, m_j \in M,\\
&\forall a \in \left\{0, 1, ..., \left(\frac{LCM(T)}{T_i} - 1\right)\right\},\\
&\forall b \in \left\{0, 1, ..., \left(\frac{LCM(T)}{T_j} - 1\right)\right\}:\\
&((m_i \neq m_j) \wedge \exists m_i^{[v_k,v_l]} \wedge \exists m_j^{[v_q,v_r]}\\
&\ \wedge (([v_k, v_l], [v_q, v_r]) \in L_{wl}) \Rightarrow\\
&((a \times T_i) + O_i^{[v_k,v_l]} \geq\\
&(b \times T_j) + O_j^{[v_q,v_r]} + T_{slot})\\
&\vee\\
&((b \times T_j) + O_j^{[v_q,v_r]} \geq\\
&(a \times T_i) + O_i^{[v_k,v_l]} + T_{slot}).
\end{aligned} \tag{8.3}$$

As already stated for Equation 8.2, it is expected that the message instances $m_{i,x}$ have an offset of one period T_i between them. The same principle applies to $m_{j,x}$ and T_j.

A specific problem that needs to be solved in an IEEE 802.11 network is the potential interference of the IEEE 802.11 beacon sent by the AP with the traffic sent by the wireless nodes. However, this is considered a non-crucial measure for performance improvement, affecting the delivery latency and jitter, and has therefore not been addressed further in this contribution.

The constraints defined up to this point protect critical traffic from collisions caused by uncoordinated transmissions between devices that are part of the same network.

8.4.3 Hardware implementation

Implementation goals and features

The goal of the developed hardware implementation is to evaluate the performance of the proposed mechanism based on selected performance metrics:

- Performance evaluation of the fault prevention in the critical traffic handling mechanism MAC_CH compared to DCF (MAC_CH_None[1]) as the default MAC mechanism in IEEE 802.11. In addition, the impact of two other MAC configuration options on performance is evaluated: default vs. prioritized channel access and active vs. inactive channel time. The evaluation and comparison is done in terms of reliability and delay for different traffic patterns, MAC mechanisms configurations and interference characteristics via the following performance metrics:
  - **Reliability**. It indicates the proportion of sent critical messages that arrive at their destination. There can be various reasons why the messages are not delivered, as the following two performance metrics show.
  - **Percentage of dropped messages**. Due to the limited memory available for the data to be transmitted on the sender nodes, the network stack of the operating system may notify the sender application that it can no longer process messages and discard them. This performance metric is used to indicate the proportion of critical messages that are discarded for this reason.
  - **Percentage of interfered messages**. Specifies the percentage of critical messages that are not received even though they have been transmitted.
  - **Delay between message arrivals**. Measures the time between the arrival of critical messages as an indication of uncertainty in the timing of the messages, with low jitter being the desired outcome.
- Performance evaluation of the clock synchronization with different traffic patterns, MAC mechanisms configurations and interference characteristics using the following performance metrics:
  - **Offset from master**. Evaluates whether the offset between the master and slave clocks is small enough to follow schedules with an acceptable slot granularity. It also checks that the offset does not suffer from sudden fluctuations that could cause the nodes to be out of synchronization and make it difficult to follow the time-triggered schedules. This evaluation is of interest for this contribution due to the particularities of the wireless channel and the fact that an out-of-the-box PTP stack is used.
  - **Percentage of in-sync time**. Indicates the percentage of time that the slave nodes are within the threshold from which they are considered synchronized with the master.

[1] In the performance evaluation, the label MAC_CH_None is used to refer to the scenarios where MAC_CH is not used.

The hardware implementation developed in this contribution has the following features:

- Implementation of wireless nodes with IEEE 802.11 as physical and data-link layer technology. The selected baseline MAC protocol from IEEE 802.11 is EDCA, coming from IEEE 802.11e. This selection does not exclude that the results are likely applicable to more recent standard updates.
- Implementation over the Atheros AR928X IEEE 802.11n chipset with PCIe interface. The chipset is a COTS hardware whose main target is consumer electronics laptops. The chipset is installed in an industrial PC, the MFN-100, which runs a non-real-time version of the Linux kernel 4.15.0. Porting to other kernel versions should be straightforward as long as there are no major changes between kernel versions in the code of the software modules that are modified in this contribution. Further, the chipset follows a soft MAC approach, where part of the MAC functionality can be controlled in software.
- The implementation modifies the open source driver for Atheros 802.11n PCI/ PCIe chips in Linux systems, `ath9k`, to enable several built-in hardware mechanisms. The aim is to control these mechanisms as close as possible to the hardware to achieve better performance. Furthermore, since it is an open source driver, it is easier to access the code compared to an implementation that uses proprietary code.
- The data to be transmitted is assigned a priority in accordance with IEEE 802.11e. The priority is defined as part of the scheduling configuration. The priorities are added to the frames in the operating system sockets. The frames are then placed in different hardware queues according to their priority.
- Modified IEEE 802.11 data-link layer, in particular the MAC layer, to include the following aspects:
  - Dispatching of critical data according to the fault prevention in the critical traffic handling mechanism `MAC_CH`. Dispatching is similar to the behavior of ST handled by the GCLs of IEEE 802.1Qbv in TSN.
  - Implementation for APs and wireless end systems in IEEE 802.11 infrastructure (BSS) and ad-hoc (IBSS) modes.
- New application layer for sending and receiving data. Allows different generation patterns: `TG_Periodic_Strict`, where the messages to be sent during a period are sent together in a burst, `TG_Periodic_Random`, where the messages are sent randomly over the period, and `TG_Sched_Sync`, where the messages are generated just before the schedule has a slot for their transmission.
- Fault-injection mechanisms. IEEE 802.11 nodes based on the Raspberry Pi 3 Model B single-board computer can generate CSMA/CA interference by using the same applications for sending data as mentioned in the previous point.
- Clock synchronization via the wireless interface is handled according to IEEE 802.1AS. For simplicity, the protocol is deployed in the application layer using `ptpd`, a PTP Linux implementation.

Implementation details

Linux IEEE 802.11 MAC driver architecture. The operation and control of the hardware in Linux is handled by device drivers in the form of kernel modules. The modules are dynamically loaded into the kernel at runtime when they are needed, typically when the hardware associated with the driver is detected. Despite the wide variety of hardware options, operating systems such as Linux aim to provide applications with uniform interfaces that hide the details of the devices behind standardized calls. This uniform interface is expressed in Linux by means of three main types of device drivers, which are classified according to the way they transfer data: Character drivers, which are accessed as a stream of bytes in a file; block drivers, where data is transferred in memory blocks; and network drivers, where the transfer is via sockets and formatted as network packets, which is an operating system's typical name for frames or messages.

As soon as a network driver is loaded into the kernel, the operating system provides a network interface for the exchange of frames and additional management functions such as the assignment of network addresses, the setting of transmission parameters or the keeping of statistics. Due to the particularities of wireless transmissions, the MAC layer of IEEE 802.11 in Linux is managed by the Media Access Control Sublayer Management Entity (MLME). MLME tries to tackle the varying reliability, security issues and power constraints in wireless networks by defining a set of operations for managing the scanning, joining, association and re-association, power management and time synchronization of the network. Depending on the hardware architecture, MLME can be done entirely in hardware (full MAC) or software (soft MAC). The latter has the advantage that it simplifies the hardware and offers more flexibility. Two kernel modules, `mac80211` and `cfg80211`, take over MLME responsibility for all IEEE 802.11-based soft MAC network devices in Linux (Figure 8.3).

A reasonable way to approach the sometimes complex nature of kernel modules and their interfaces is to visualize them in terms of the paths used by the transmissions when sending and receiving, and how the configuration is done. In the depicted driver architecture, there are five main paths: for sending and receiving data, for sending and receiving beacon frames, and for configuration. In this implementation, the most relevant path is the one used to send frames, as it deals with TT/ST handling. When a socket application wants to send a frame, a function of the interface `net_device_ops` is called via the `mac80211` module. This module is responsible for creating the IEEE 802.11 frame header including the QoS field, updating the statistics, segmenting the frames if they are larger than permitted by IEEE 802.11 and queuing them if the hardware queues are not available. Due to the limited memory on the hardware, if the applications attempt to send more data than can be processed by the hardware, the hardware may have to notify the Linux module to stop sending data for a while. When a frame is ready for transmission, a call from the `ieee80211_ops` interface hands it over to the specific hardware modules that are addressed in the next section. `Mac80211` not only deals with the frames sent, but is also involved in the process of receiving these frames.

Enabling scheduled traffic in the Atheros AR928X chipset. The implementation is based on the Atheros AR928X chipset, an IEEE 802.11n-compliant solution

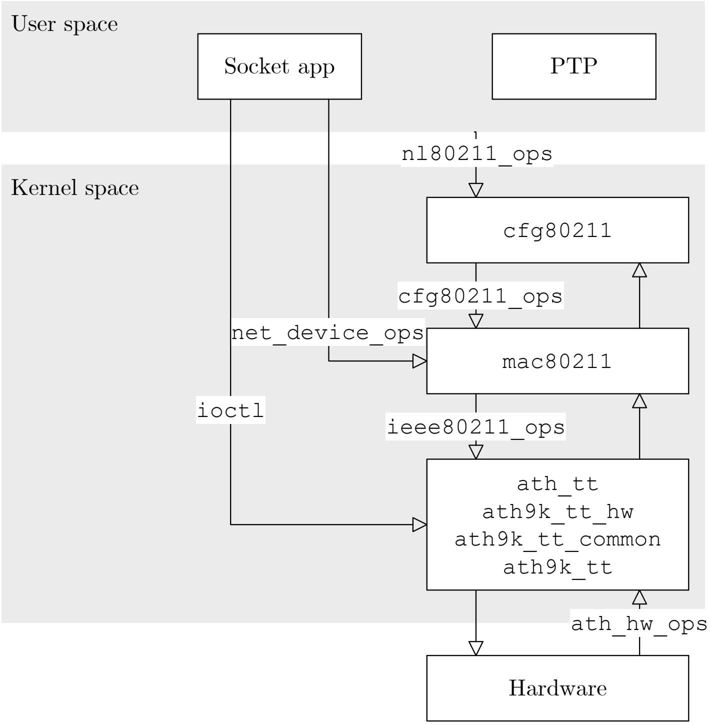


Figure 8.3: Fault prevention in the critical-traffic handling: Architecture and interfaces of the Linux IEEE 802.11 MAC and the modified `ath9k` driver modules

in the 2.4 GHz band that comes with throughput of up to 150 Mbps. The chipset covers the physical and MAC layers, both of which are accessible from the host device via a series of registers for configuration and status messages. For these purposes, the `ath9k` driver is used as its code is available as open source in the Linux kernel. The modifications made in the implementation described here relate to the MAC layer and are therefore independent of whether the device is an AP or not. An implementation of the proposal using other hardware models is theoretically possible, provided that a mechanism is in place to control the exact timing for triggering the transmission of data frames.

The chipset MAC offers ten transmission queues (Figure 8.4). The assignment to the queues is configurable, but by default data frames are assigned to queues Q0 to Q3 based on their EDCA priority, which is specified as part of the frame header. Queues Q4 to Q7 remain unused, but can also be used for data frames if required, so that the entire range of EDCA priorities is covered. Finally, queues Q8 and Q9 are reserved for beacon frames. For frames transmitted over both IEEE 802.11 and Ethernet networks, the EDCA priority value is often chosen to match the priority of the VLAN tag. The partition of traffic into different queues based on their priority effectively prevents higher priority traffic from interfering with lower priority traffic in the device memory. Each transmission queue consists of two modules that jointly manage the frame dispatch: The queue control unit (QCU), which selects the frames to be transmitted based on the so-called QCU frame scheduling policy, and the DCF unit (DCU), which performs the EDCA channel access.

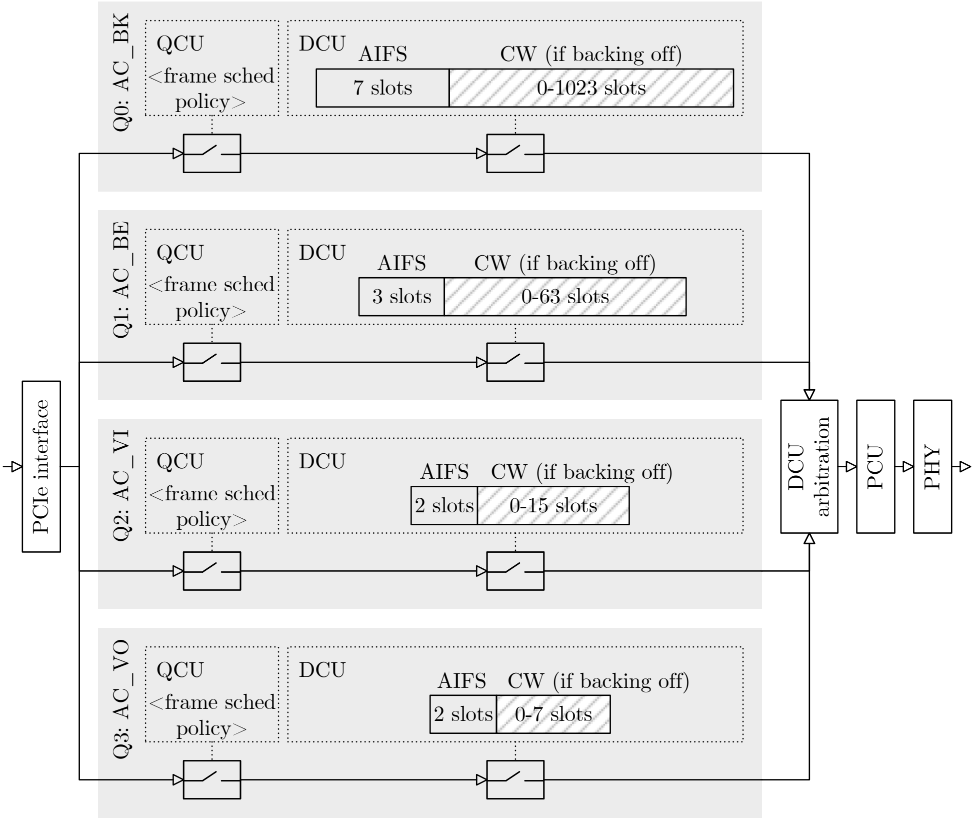


Figure 8.4: Fault prevention in the critical-traffic handling: Atheros chipset MAC queues. Only queues Q0-Q3 are shown, which correspond to the default chipset mapping between ACs and EDCA priorities.

The interface between the chipset and the host is based on PCIe. The interface is used to access the registers and perform the frame exchange. The first version of the driver was developed using a chipset with USB access, but the delay in tasks such as writing registers was in the order of ms, while the PCIe interface offers delays in the order of µs. After the frame is sent via the PCIe interface, it is stored in the chipset in a buffer in the form of a FIFO linked list. The responsibility of the corresponding QCU is to select the frames from the linked list and make them available to the DCU so that it can continue with the transfer.

The decision when to select a frame for transmission depends on the QCU frame scheduling policy. With the exception of beacon queues, which periodically trigger the beacon frame, all other queues apply an as-soon-as-possible policy by default, whereby frames are sent to the DCU as soon as there is no other frame in the DCU. Of the other methods available, the CBR method most closely resembles time-triggered scheduling as enabled by the TGs. CBR allows the selection of frames from a TG periodically, but is unfortunately not entirely useful as the TG mechanism is not limited to opening or closing the gates periodically but at an arbitrary time. One option would be to change the value of the CBR period each time a frame needs to be sent, but tests showed that the hardware did not cope well with dynamically changing the period and frames were not sent on time. Fortunately, the chipset offers a solution that makes it possible to determine the time at which a queue is allowed to transmit data. The solution is based on the one-shot mechanism (Figure 8.5), which is used in combination with the CBR QCU frame scheduling policy. First, the user shall write in a register which QCUs shall select frames for transmission. Then the one-shot mechanism is used to trigger a frame transfer from each active QCU to the corresponding DCU.

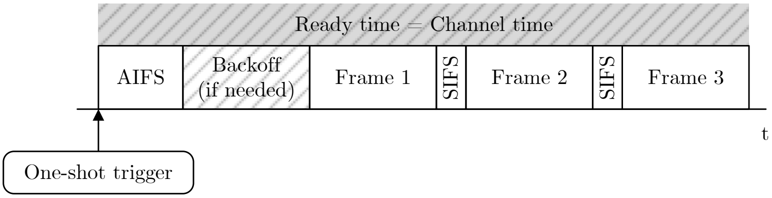


Figure 8.5: Fault prevention in the critical-traffic handling: Example of the transmission of three frames, and the use of one-shot, ready-time and channel-time mechanisms to access the channel.

In addition, the chipset's QCU frame scheduling policies allow the transmission of a single frame instead of a series of frames as in the TGs or TXOP mechanisms of TSN or IEEE 802.11, respectively. This behavior can be modified by the chipset's ready-time mechanism. Ready time specifies the duration during which the QCU marks the frames as ready to be sent to the DCU, emulating the effect of the TG being open for a user-defined duration.

Once the QCU provides the frame to the DCU, the EDCA channel access protocol is followed. The CW values can be adjusted accordingly to avoid the problem of overlapping priorities with the default AC parameters. In this regard, the chipset's channel-time mechanism only requires the first frame in a user-defined time period to arbitrate for channel access. For the remaining frames, waiting for AIFS and CW is avoided, leaving only the SIFS in between (Figure 8.5). The lack of continuous arbitration efficiently reduces the overhead of DCU channel access for each frame. Moreover, the duration of the channel time can be set to be equal to the ready time, allowing the scheduler to accommodate as many frames as possible within the ready-time duration. After the DCU grants a frame access to the medium, it is handed over to the protocol control unit (PCU), which is responsible for sending the frame to the baseband logic and performing other DCF-compliant tasks.

The kernel modules `ath_tt`, `ath9k_tt_hw`, `ath9k_tt_common` and `ath9k_tt` depicted in Figure 8.3 are the modified version in this contribution of the `ath9k` driver modules provided in the Linux kernel. These modules are responsible for setting the proper configuration values for the described hardware mechanisms and for transferring frames from the mac80211 module to the chipset and vice versa via the PCIe interface.

To enable the TGs mechanism, a schedule in the format of a GCL shall be configured in the ath9k kernel modules mentioned above. The schedule is made available to the kernel modules by exposing them as character drivers, which enables a direct path from the user applications via the so-called ioctl calls of the operating system. The schedule configuration is distributed and applied before network operation. However, the driver can dynamically load a new schedule so that the configuration mechanisms from the IEEE 802.1Qcc "Stream Reservation Protocol Enhancements and Performance Improvement" of TSN could be used in the future.

When a schedule is available in the kernel, the times for opening and closing the TGs are programmed using Linux kernel timers. When the timers expire, their handling routines reconfigure the one-shot, ready-time and channel-time mechanisms. The timer expiration is handled as an interrupt in the operating system. In the interrupt-handling context certain kernel operations are restricted and processing should be as short as possible so as not to impair performance of the entire system. This restricts operations to write registers as required to reconfigure the chipset mechanisms that apply the TG states. Fortunately, the Linux kernel provides the work queues mechanism to defer the handling of very frequent and intensive operations out of the interrupt context. As soon as the interrupt occurs, the interrupt handler briefly places a work item in the work queue, waiting to be processed as quickly as possible. When being processed, the work takes over the reconfiguration of the chipset mechanisms mentioned above.

After running some experiments on forcing the TG state using interrupts and work queues, it was found that interrupts and work queues were precise enough to apply a new TG state with a frequency of 1 ms when using the PCIe interface to the chipset. Smaller values seemed to lead to errors in the driver. Unfortunately, it was also observed that queuing works into the work queue at a high frequency caused some works to be lost. Although this was rare, some countermeasures were taken when programming the driver so that the loss of some works did not cause the schedules between the nodes to be out of synchronization.

As for the mapping of transmissions into priorities, clock synchronization messages are sent as low-priority traffic because the prioritization is generally not considered critical to the protocol [122]. Data transmissions have a medium priority. In addition, the wireless network periodically sends out beacon frames to announce the network, among other purposes. These frames are transmitted with the highest priority and have an impact on data transmissions. This interfering effect has not been thoroughly investigated and addressed, leaving this is as a future work.

Clock synchronization based on PTP. The option selected for the synchronization between the various Linux end systems is the PTP software ptpd, which runs as a user-space application. As previously mentioned, PTP is part of IEEE 1588, a superset of IEEE 802.1AS. Further details about PTP can be found in the TSN section (Section 5.2.1). Ptpd is selected due to the simplicity of running the clock synchronization as a readily available application in the user space. This out-of-the-box solution avoids additional driver modifications. In addition, the evaluation of a clock synchronization done in software and over wireless medium provides insights into how good synchronization can be under such non-ideal conditions and whether it can be used to schedule time-triggered traffic.

In the selected configuration for `ptpd`, the AP is selected as the PTP master, while the end systems act as slaves. In the test scenario, it is possible to use an AP because the AP acts as an end system, runs the same software and counts with a clock. The master triggers the synchronization process with the slaves based on a user-defined synchronization interval. Frequent synchronization rounds help to reduce the potential drifts between the clocks of the different nodes, but cause a significant overhead. However, as previously mentioned the clock synchronization messages are sent with low priority so that they do not interfere with critical data.

The experiments, which are explained in detail later in the performance evaluation, have shown that the software solution without timestamping support and working over the wireless medium results in the offset from the master being subject to sudden and frequent variations. To reduce the susceptibility to such variations, `ptpd` enables the processing of synchronization information via filters. The filters are applied to a user-defined window of values collected in the previous synchronization rounds. A median filter is chosen, which selects the intermediate values used to calculate the offset from the master and omits the extreme values. In this way, sudden variations in the values do not lead to dramatic changes unless the trend is kept over time. Once the offset from the master is calculated, `ptpd` is configured to adjust the clock rate to minimize such an offset instead of making sudden corrections.

Timing guarantees. As previously mentioned in the real-time guarantees in the layered architecture (Section 4.1.3), each of the steps involved in the communication process shall either be scheduled or have sufficient time so that there is a high probability that one step will be completed before the next. The modified driver adds the ability to schedule when frames are triggered by the hardware queues according to a GCL generated by the scheduler. The scheduler should have generated the GCL according to the constraints previously described in the scheduling section, which means that aspects of the data to be transferred such as its data size, occurrence pattern and delivery deadline are taken into account. By the time the GCL selects a frame for transmission, the frame should hopefully have been placed in the corresponding queues, having made its way from the sending application, through the kernel modules, to the hardware. Experiments have shown that the entire process can be completed at a maximum pace of 1 ms, with results showing that a higher frequency leads to data loss.

On the receiving side, the data is made available to the end system after it has been transmitted from another end system under the control of the TG mechanism. This means that reception is also scheduled. Once the transmission has arrived at the receiving end, the operating system must make it available to the receiving application, a final step that should not take long to complete.

8.4.4 Performance evaluation

The purpose of the evaluation is to assess the MAC mechanism MAC_CH realized by the application of ST over IEEE 802.11 using a hardware implementation. A comparison of the performance between the MAC with ST and the standard IEEE 802.11 DCF (MAC_CH_None) is carried out. In addition, the impact of two other MAC configuration options on performance is evaluated: default vs. prioritized channel access and active vs. inactive channel time. The performance measures considered are reliability, delay between message arrivals and clock synchronization precision.

Scenario description

The Atheros AR928X IEEE 802.11 chipset used in the implementation is installed via a PCIe interface on five MFN-100 industrial PCs running a non-real-time version of Linux and serving as a proof of concept for the deployment of industrial equipment. The five nodes, which are connected in an infrastructure topology (Figure 8.6), are placed evenly over an area of 150 m^2 in an office building. Three additional Raspberry Pi nodes are used to generate interference, two of which act as senders and one as a receiver.

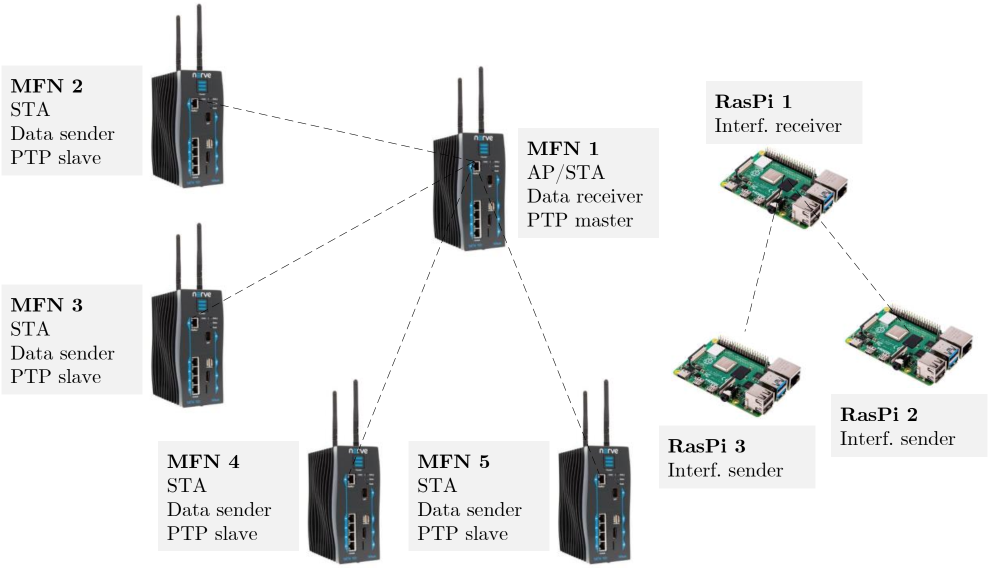


Figure 8.6: Fault prevention in the critical-traffic handling: evaluation network

The performance of the implementation is evaluated in scenarios resulting from the combination of different protocol-related, traffic-related and fault-injection-related parameters.

The protocol-related parameters, summarized in Table 8.7, include the MAC protocol used, either DCF or ST, whether default or prioritized channel access parameters apply, and finally whether the channel time is active. Table 8.7 also describes the assignment of the different kinds of transmissions to the priorities. The ST schedule is realized with a GCL structure in two segments that repeat over time. One segment is used for medium-priority transmissions, while the other segment is responsible for low-priority transmissions. The segments are separated from each other by a time buffer, so that they are less likely to overlap. Retransmissions via ARQ have been deactivated so that each message only has one chance of being delivered. Features like fragmentation and the handling of hidden nodes are also deactivated, as they are of little relevance for the experiments. Rate adaptation is enabled, since the experiments have shown that setting the rate to a fixed value corresponding to the maximum rate R causes too many message errors. The nodes are synchronized with `ptpd`, with one node statically selected as the PTP master. `Ptpd` runs as a user-space application on the nodes and does not provide support for hardware timestamping in this evaluation. In addition, timestamping is performed in a two-step process. The evaluation of the clock synchronization provides information on how good the synchronization can be under such non-ideal conditions and whether it can be used to enable ST. The synchronization interval for PTP is set to 0.25 s, while the synchronization is interpreted using a median filter with the last 128 values, the largest possible in `ptpd`, to protect the synchronization protocol from sudden changes.

Table 8.7: Fault prevention in the critical-traffic handling: protocol-related evaluation parameters

Parameter	**Value**
MAC protocol	DCF (`MAC_CH_None`)
	ST (`MAC_CH`)
Channel access	Default node: • AIFS = 2 slots, $T_{AIFS} = T_{SIFS} + 2T_{ST} = 28\,\mu$s (IEEE 802.11g) • CW_{min} = 15 slots, Minimal $T_{CW} = [0, 15]T_{ST} = [0, 135]\mu s$ (IEEE 802.11g) • CW_{max} = 1023 slots, Maximal $T_{CW} = [0, 1023]T_{ST} = [0, 9207]\mu s$ (IEEE 802.11g)
	Prioritized node: • AIFS = 0 slots, $T_{AIFS} = 0\,\mu$s (IEEE 802.11g) • CW = 0 slots, $T_{CW} = 0\,\mu$s (IEEE 802.11g)
Channel time (only for the ST MAC protocol)	Inactive
	Active and equal to the duration of the current ST slot

Parameter	Value
DCF and ARQ characteristics	• $T_{ST} = 9\,\mu$s (IEEE 802.11g) • $T_{SIFS} = 10\,\mu$s (IEEE 802.11g) • See channel access row above for AIFS and CW characteristics. Note that even if MAC_CH does not require them, they are among the configurations to be tested. • RTS/CTS: disabled • ACK frames: enabled (since it could not be disabled in the hardware) • ARQ: disabled • Refer to Table 8.6 for additional details.
Priority mapping	• High priority: beacon frame • Medium priority: generated data • Low priority: network services and clock synchronization
ST schedule (GCL)	10 ms long, with two segments triggering medium- and low-priority data, respectively, and a 1 ms buffer between the two.
Fragmentation	Disabled
Rate adaptation	Enabled
Maximum rate R [Mbps]	54 (IEEE 802.11g)
Clock synchronization protocol	PTP (ptpd software)
PTP synchronization timestamping	Software-only two-step process
PTP synchronization interval [s]	0.25
PTP synchronization filter	Median filter with the last 128 values
Beacon frame period [ms]	100
WiFi security options	WPA

The traffic-related evaluation parameters, which are summarized in Table 8.8, cover the evaluation of different message sizes $m_i.size$, which are classified into small, medium and large. Different levels of utilization of the available bandwidth are also taken into account, divided into low and mid-high utilization, resulting in different burst sizes. At least 10000 data messages are generated for each test. Another parameter that applies to all scenarios refers to the use of non-blocking socket calls in Linux, causing the application to drop the data if transfer requests flood the hardware.

Finally, the fault-injection evaluation parameters, summarized in Table 8.9, indicate which type of interference occurs during the evaluation, namely either the default office background noise present in the testing environment or the office noise complemented by the additional disturbance generated by some Raspberry Pis.

Table 8.8: Fault prevention in the critical-traffic handling: traffic-related evaluation parameters

Parameter	**Value**
Message size $m_i.size$ [bytes] Estimated utilization [%] (burst size)	Small: 32 Low: 1.1% (1x)
	Small: 32 Mid-high: 28.6% (26x)
	Medium: 512 Low: 1.8% (1x)
	Medium: 512 Mid-high: 29.1% (16x)
	Large: 2048 Low: 4.1% (1x)
	Large: 2048 Mid-high: 28.4% (7x)
Number of generated messages $\|M\|$	$\geq$ 10000
Message generation period T_i [ms]	50
Generation pattern	Random over the period
Addressing	Unicast, with one hop to the AP
Linux socket calls	Non-blocking

Table 8.9: Fault prevention in the critical-traffic handling: fault-injection-related evaluation parameters

Parameter	**Value**
Type of interference	Office noise
	Office noise + additional disturbance
Additional disturbance characteristics	Data size [bytes]: 256 Burst size: 20x $m_i.size$ Period [ms]: 100 Number of generating nodes: 2x Estimated utilization: 5.7%

A total of 72 configurations/scenarios are evaluated, which result from the combination of various protocol-related (MAC protocol, channel access and channel time), traffic-related (message size and estimated utilization) and fault-injection-related parameters (type of interference).

Results

Table 8.10 shows the reliability results for each protocol, traffic and fault-injection parameter obtained after aggregating the results from all scenarios in which the given parameter is present. The results are also provided when considering only the parameter options that lead to the best and worst scenarios, which are described in detail in the following analysis. For each option, the proportion of messages delivered (del.) is given. A distinction is then made between the messages that were dropped due to overload before the transmission attempt (drp.) and the messages that were sent but interfered (int.).

Considering all scenarios, the ST-based MAC protocol performs better than DCF, default channel access parameters are generally better than prioritized parameters, and activating the channel timing feature is more beneficial than disabling it. Furthermore, scenarios with a high-medium utilization tend to show poorer performance due to a more congested medium, which also applies to scenarios where an additional interference occurs. The analysis of the results shows that the best scenarios are those in which the coincidence of DCF as MAC protocol, prioritized channel access and a mid-high medium utilization does not occur. Under these favorable conditions, values above 0.95 reliability are easy to achieve, and DCF and ST deliver similar results. The scenarios with the worst results, which are the complement to the scenarios with the best results, have reliability values of less than 0.4 and even as low as 0.1.

Some conclusions can be drawn from the reliability results. First, the MAC protocols DCF and ST do not provide divergent results unless other parameters such as prioritized channel access are considered, which is detrimental to DCF and should not be used in combination. The reason is that with DCF, multiple prioritized nodes might try to access the medium at the same time, which leads to collisions. Secondly, the results also confirm the obvious fact that a more congested medium, either due to mid-high utilization from traffic generated in the network or due to additional interference, reduces reliability. Therefore, it is important to emphasize that transmission scheduling, as in ST, does not per se lead to better reliability. A large part of the reliability improvement comes from the admission control mechanisms. By default, DCF is not used in combination with admission control, and nodes are not aware of the medium usage quota they must not exceed to avoid channel capacity overflow. Such DCF nodes try to send data as soon as they have some data available for exchange, which could affect the reliability of transmissions. Finally, the percentage of dropped and interfered messages increases with mid-high utilization and additional interference. Such a scenario leads to congestion on the wireless channel and transfers the problem to the network interface, which is unable to accept further transmission requests from the application, so the messages are discarded before they get a chance to be sent.

Table 8.11 summarizes the results for the delay between message arrivals. The results are again given for each configuration parameter after aggregating all the scenarios in which the respective parameter is present. The results for the parameters that lead to the best and worst case scenarios, which are explained in more detail in the following analysis, are also included. In all cases, the message size and utilization parameters are not considered as they are not of interest, e.g., bursty scenarios inherently have a shorter delay between arrivals than scenarios where messages are sent one

Table 8.10: Fault prevention in the critical-traffic handling: results on reliability

Parameter		**Results all scenarios**			**Results best scenarios**			**Results worst scenarios**		
		Del.	**Drp.**	**Int.**	**Del.**	**Drp.**	**Int.**	**Del.**	**Drp.**	**Int.**
MAC protocol	DCF	0.7868	0.0673	0.1459	0.9763	0.0137	0.0100	0.2185	0.2281	0.5534
	ST	0.9582	0.0025	0.0393	0.9582	0.0025	0.0393	-	-	-
Channel access	Default	0.9347	0.0102	0.0551	0.9347	0.0102	0.0551	-	-	-
	Prioritized	0.8675	0.0380	0.0945	0.9973	0.0000	0.0027	0.2185	0.2281	0.5534
Channel time	Inactive	0.8630	0.0357	0.1013	0.9550	0.0082	0.0367	0.2185	0.2281	0.5534
	Active (ST only)	0.9774	0.0009	0.0218	0.9774	0.0009	0.0218	-	-	-
Message size, utilization	Small, low	0.9983	0.0000	0.0017	0.9983	0.0000	0.0017	-	-	-
	Small, mid-high	0.8431	0.0318	0.1251	0.9395	0.0056	0.0549	0.3612	0.1632	0.4757
	Medium, low	0.9984	0.0000	0.0016	0.9984	0.0000	0.0016	-	-	-
	Medium, mid-high	0.8247	0.0341	0.1412	0.9511	0.0025	0.0464	0.1923	0.1923	0.6154
	Large, low	0.9980	0.0000	0.0020	0.9980	0.0000	0.0020	-	-	-
	Large, mid-high	0.7442	0.0785	0.1773	0.8726	0.0285	0.0989	0.1020	0.3287	0.5692
Type of interference	Office noise	0.9223	0.0265	0.0513	0.9874	0.0074	0.0052	0.2059	0.2359	0.5582
	Office noise + additional disturbance	0.8799	0.0217	0.0984	0.9389	0.0036	0.0574	0.2311	0.2202	0.5487

at a time. Therefore, the interest is to evaluate whether the remaining parameters cause the delay between arrivals to differ greatly between message instances.

Table 8.11: Fault prevention in the critical-traffic handling: results on the delay between message arrivals

Parameter		**σ[s] results scenarios**		
		All	**Best**	**Worst**
MAC protocol	DCF	0.0790	0.0196	0.1383
	ST	0.0189	0.0189	-
Channel access	Default	0.0200	0.0200	-
	Prioritized	0.0579	0.0176	0.1383
Channel time	Inactive	0.0491	0.0194	0.1383
	Active (ST)	0.0186	0.0186	-
Type of interference	Office noise	0.0382	0.0187	0.1362
	Office noise + add.	0.0396	0.0195	0.1405

One of the desirable and confirmed results of the ST MAC protocol is counting with fewer deviations in the delay between message arrivals, i.e., jitter, than with DCF. Similarly, an active channel time mechanism reduces the time variance between arrivals by eliminating the need to sense the channel for the subsequent messages in a burst. Surprisingly, prioritized channel access does not lead to a lower variance between message arrivals. This behavior is probably due to the higher number of messages that are lost when combining DCF and prioritized channel access. No significant differences are observed for the different interference types. The results show that the scenarios in which the combination of DCF and prioritized channel access does not occur provide the best outcome. The difference between the best and the worst result is of one order of magnitude. Further, the use of ST is expected to improve the delivery latency in the scenarios where the applications that generate and consume the data are synchronized with the ST schedule. In such cases, it may be possible to align the steps of data generation, transmission and consumption to minimize end-to-end latencies.

Finally, Table 8.12 presents the results of the clock synchronization quality for the message size and utilization, and type of interference parameters, as these two parameter groups have the greatest influence on clock synchronization. The outcome is provided after aggregating the results of all scenarios in which the parameter in question is present and also filtering to have the parameters, which are clarified below, leading to the best and worst scenarios. The aim is to have a synchronization offset of at most ±1 ms in order to be able to follow the ST schedules.

Table 8.12: Fault prevention in the critical-traffic handling: results on the quality of clock synchronization

Parameter		Results all scenarios			Results best scenarios			Results worst scenarios		
		$\overline{\mathbf{X}}$ [s]	**σ [s]**	**In sync**	$\overline{\mathbf{X}}$ [s]	**σ [s]**	**In sync**	$\overline{\mathbf{X}}$ [s]	**σ [s]**	**In sync**
Message size, utilization	Small, low	0.0000	0.0006	0.9094	0.0000	0.0006	0.9094	-	-	-
	Small, mid-high	0.0000	0.0014	0.7483	0.0003	0.0013	0.7841	-0.0003	0.0016	0.7126
	Medium, low	0.0001	0.0007	0.9005	0.0001	0.0007	0.9005	-	-	-
	Medium, mid-high	-0.0001	0.0014	0.8007	-0.0002	0.0009	0.9065	-0.0001	0.0019	0.6948
	Large, low	0.0000	0.0008	0.8617	0.0000	0.0008	0.8617	-	-	-
	Large, mid-high	-0.0001	0.0018	0.7988	0.0000	0.0013	0.8874	-0.0003	0.0023	0.7103
Type of interference	Office noise	0.0001	0.0008	0.9103	0.0001	0.0008	0.9103	-	-	-
	Office noise + additional disturbance	-0.0001	0.0014	0.7629	0.0000	0.0009	0.8199	-0.0002	0.0019	0.7059

The mean value of the offset values $\overline{X}$ is given to show that the offset variations are happening around zero as expected. The standard deviation σ around this mean value is lower than 1 ms for scenarios with low utilization and only office interference, which shows that higher utilization and more interference strongly influence clock synchronization. In the best configurations, i.e., without the higher medium congestion, the nodes are still out of sync at least 10% of the time, which shows how vulnerable software-based wireless synchronization based on PTP can be. However, even if the nodes are out of sync for some time, this does not seem to hinder the ability of ST schedules to be followed and achieve better results in terms of reliability and delay jitter than DCF, as shown in Table 8.10. Therefore, a purely software-based clock synchronization protocol might be sufficient in systems without tight timing requirements.

8.4.5 Conclusions

This contribution presents the fault-prevention mechanism `MAC_CH` for handling critical traffic based on the allocation of time-slots to transmissions considering their timing requirements. The proposed solution is based on the integration of wired and wireless technologies, Ethernet and IEEE 802.11, respectively. The scheduling solution extends the scheduler for TSN and TTE traffic that is already available on the wired segments. As TSN mechanisms such as ST are increasingly used in real-time wired applications based on Ethernet, an implementation is presented where transmissions take place in time slots using ST over IEEE 802.11 to evaluate the performance of the wireless segment. The implementation relies on IEEE 802.11 COTS to enable TSN's ST using existing hardware mechanisms configured via a customized Linux driver version. Several MAC protocol configurations are tested against different traffic patterns and interference levels. The results show that the reliability can be strongly influenced by the MAC protocol configuration, but in general the introduction of ST does not directly lead to a reliability performance improvement compared to the regular DCF mechanism of IEEE 802.11. The main merit lies in the admission control mechanism, which is often used together with mechanisms such as ST, as it prevents a congested medium from leading to poor reliability. The results also show that ST improves DCF in terms of delivery jitter. Finally, PTP implemented in software is used to synchronize the network. The performance results show that a software-only version is quite sensitive to an overloaded medium, but is still good enough to support the ST mechanism even during the short periods when synchronization is lost.

In addition to reliability and timing guarantees, availability and safety are other requirements on which `MAC_CH` could have a positive impact. Determinism depends on the conditions for which it is defined, but should also benefit from a higher proportion of successful transfers. Scheduling critical transmissions is not expected to incur additional overhead or consume part of the throughput, except for the required clock synchronization protocol. In contrast, this solution is more complex than standard DCF and has higher costs, but since it is based on COTS devices, it should be less costly than other dependable and real-time solutions from OT. On the downside, the use of offline schedules for handling critical traffic does not favor adaptability.

8.5 Fault prevention in the best-effort handling (MAC_BH_1, MAC_BH_2 and MAC_BH_3)

8.5.1 Mechanisms description

The remaining time in the schedule after the allocation of TT/ST and RC/CBS traffic following MAC_CH is for BE traffic. In the case of BE traffic, no guarantees are provided by TTE or TSN. However, TTE and TSN are designed to operate with a wired technology such as Ethernet that provides full-duplex connections between pairs of devices on the network. In contrast, the broadcast nature of the wireless medium creates collision domains that might contain a large number of nodes. This, and the fact that a significant proportion of slots may already be allocated to TT/ST and RC/CBS traffic, could significantly affect the ability to carry BE traffic over the wireless segments. Therefore, three different options for handling BE traffic are proposed [2][3] to improve their performance and reduce the presence of faults (as initially indicated in Table 7.8 and Table 7.9 in the fault-handling proposals). The generated schedules, which take into account the handling of all traffic classes, are distributed to the participant nodes before the network is put into operation. Further details on scheduling can be found later in Section 8.5.2.

Pre-scheduled time slots (MAC_BH_1)

The time available for BE traffic is divided into time slots that are pre-assigned to nodes using a round-robin scheduling policy (Figure 8.7). Such a slot assignment favors fairness when the characteristics of the BE traffic are not modeled. The approach is comparable to the contention-free phases of various protocols, e.g., FlexRay, PROFINET, POWERLINK, Bluetooth, IEEE 802.15.4 or WirelessHART.

The pre-scheduled time slots mechanism for BE traffic is described by the pseudocode in Algorithm 8.2. The procedures from the pseudocode are clarified in Appendix A. In the pseudocode it can be appreciated how slots are assigned to critical traffic (line 4) or BE traffic (line 8).

Algorithm 8.2 Fault prevention in the best-effort handling: Pseudocode describing the MAC mechanism MAC_BH_1.

```
 1: procedure MAC_BH_1
 2:     while true do
 3:         wait_for_assigned_slot()
 4:         if is_current_slot_critical() then
 5:             if not is_critical_queue_empty() then
 6:                 m ←dequeue_critical()
 7:                 send_asap(m)
 8:         else if is_current_slot_be() then
 9:             if not is_be_queue_empty() then
10:                 m ←dequeue_be()
11:                 send_asap(m)
```

Since there is no further protocol overhead in this mechanism, each slot T_{slot} should

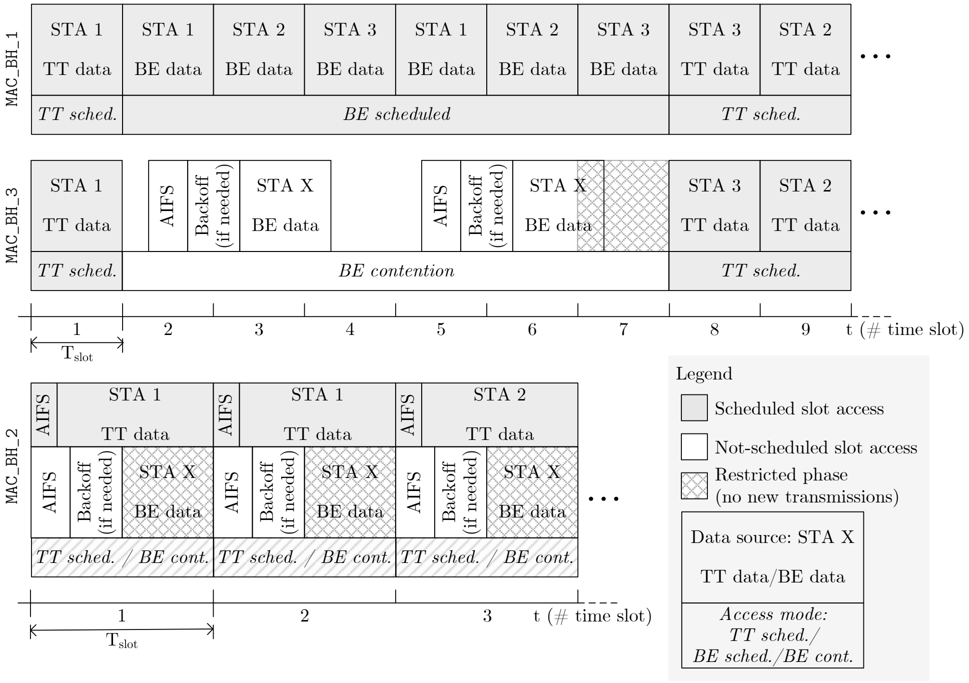


Figure 8.7: Fault prevention in the best-effort handling: example of time allocation

be long enough to accommodate one transmission T_{data} with a short guard time, which is chosen to be the slot time T_{ST}. If a node wants to send a BE message, it must wait until the BE slot assigned in a round-robin fashion comes. The time slots reserved for BE traffic are subject to the prior critical traffic allocation, but it is expected that in most cases they are spread over the duration of the schedule instead of concentrated, generally resulting in a shorter delay until access is granted. Given a set of slots reserved for critical traffic S, a certain number of wireless nodes $|V_{wl}|$, and the transmission time for a data message T_{data}, then the worst channel access delay is bounded and equal to $T_{delay} = (S + |V_{wl}| - 1)(T_{ST} + T_{data})$. The best channel access delay is almost zero, as it only corresponds to the guard interval, i.e., $T_{delay} = T_{ST}$.

Contention-based time slots (MAC_BH_2)

In MAC_BH_1, a time slot remains unused if a node has nothing to send when its assigned slot arrives (Figure 8.7). To avoid wasting bandwidth, MAC_BH_2 proposes that if a node does not use its assigned slot, the other nodes can try to gain access via a contention process as long as they have something to send. Such behavior is similar to that of slotted ALOHA, with the exception that a single node is prioritized over others in the contribution described here.

The implementation is based on all nodes performing channel sensing, but one prioritized node uses a shorter T_{AIFS} than the rest. If the prioritized node has nothing to send, the other nodes notice that the channel is idle since they sense the channel for a larger T_{AIFS}. To reduce the probability of collisions, the non-prioritized nodes wait an additional random time T_{CW}, i.e., a backoff time, selected from a CW.

Algorithm 8.3 contains the pseudocode that describes the handling of BE traffic with contention-based time slots. The procedures from the pseudocode are clarified in Appendix A. Some aspects of pseudocode to emphasize are: When a node does not use the slot assigned to it, other nodes can try to gain access via a contention process (lines 10-11), how the prioritized node uses a shorter T_{AIFS} than the rest (lines 9 and 11) and how non-prioritized nodes wait an additional random time T_{CW} (line 26).

Algorithm 8.3 Fault prevention in the best-effort handling: Pseudocode describing the MAC mechanisms `MAC_BH_2`, `MAC_BH_3b`, and the procedures for sending data following CSMA/CA (send_csma_ca) and for sending BE data (send_be).

```
 1: procedure MAC_BH_2_AND_3B
 2:     $aifs\_prio \leftarrow AIFS\_AC\_VO$
 3:     $aifs\_be \leftarrow AIFS\_AC\_BE$
 4:     while true do
 5:         wait_for_assigned_slot()
 6:         if is_current_slot_critical() then
 7:             if not is_critical_queue_empty() then
 8:                 $m \leftarrow$dequeue_critical()
 9:                 send_csma_ca($m$, aifs_prio, 0)
10:             else if not is_be_queue_empty() then
11:                 send_be(aifs_be, 0)
12:         else if is_current_slot_be() and not is_restricted_phase() then
13:             if not is_be_queue_empty() then
14:                 send_be(aifs_be, 0)

15: procedure SEND_CSMA_CA($m$, aifs, backoff_time)
16:     if not sense_channel(aifs) then
17:         return 0
18:     if backoff_time $\neq 0$ then
19:         if not sense_channel(backoff_time) then
20:             backoff_time $\leftarrow$update_backoff(backoff_time)
21:             return backoff_time
22:     send_asap($m$)
23:     return 0

24: procedure SEND_BE(aifs, backoff_time)
25:     if backoff_time $= 0$ then
26:         backoff_time $\leftarrow$rand_inside_cw()
27:     $m \leftarrow$dequeue_be()
28:     backoff_time $\leftarrow$send_csma_ca($m$, aifs, backoff_time)
29:     return backoff_time
```

The slot size T_{slot} in this mechanism must be large enough to accommodate the transmission T_{data} and the largest T_{AIFS} plus a guard interval between the transmissions for which the slot time T_{ST} is selected. Alternatively, the slot can be made even larger so that more values from the CW fit, further reducing the likelihood of the slot remaining empty. Note that the size of the CW does not increase after each collision so as not to extend the waiting times.

The shortest possible slot, labeled as the mechanism variant MAC_BH_2a, only allows the transmission from the node that has prioritized access or from any other node that happens to have a backoff time value of zero. Note that multiple non-prioritized nodes can randomize the same backoff time value, so this variant increases the likelihood of collisions. If the slot is expanded, a variant labeled MAC_BH_2b, more values from the CW will fit and therefore the non-prioritized nodes will be given more chances, although fewer slots can be allocated overall.

For MAC_BH_2, the worst channel access delay becomes $T_{delay} = (S + |V_{wl}| - 1)(T_{AIFS} + T_{CW} + T_{data})$, since in the worst case BE data can only be transmitted in the prioritized slot. The best case for BE data implies a delay of $T_{delay} = T_{AIFS}$.

Contention-based phases (MAC_BH_3)

In MAC_BH_2, if two or more BE slots occur consecutively, the channel sensing must be restarted, ignoring previous information about the state of the channel that could contribute to more efficient access. MAC_BH_3 proposes that consecutive BE slots are merged into a continuous phase, i.e., a larger slot, in which the nodes access the medium to perform BE data exchanges via contention without pre-assigned priority (Figure 8.7). The most convenient ratio between the contention-based and scheduled phases can be determined using schedulability analysis [123]. The proposed mechanism is similar to the contention phases of FlexRay, IEEE 802.15.4, WirelessHART or ISA100.11a. However, unlike these examples, the proposed mechanism could have multiple contention phases in the hyper period depending on the time remaining after the critical traffic is scheduled.

The two extreme cases for the placement of contention-based phases are, on the one hand, that all critical traffic time slots are combined in one long scheduled phase and the rest of the hyper period is left to the contention-based phase. On the other hand, the exchange of critical traffic data may be distributed evenly over the hyper period, leaving some spaces in between for the contention-based phases. The advantage of distributing the scheduled phases evenly over the hyper period is that the best-case delay in channel access for BE traffic can be reduced. The disadvantage is that the gaps between the scheduled phases may not be large enough to accommodate the time for contention and transmission. Further, if the contention-based phases are large enough, more than one transmission could fit.

When ending a contention-based phase, two alternatives are explored. Either the current backoff time counters are retained and resumed at the beginning of the next phase, an approach labeled as MAC_BH_3a, or to randomize a new backoff time value at the beginning of each contention phase, denoted MAC_BH_3b. It is assumed that the first option works better because the value of the backoff time depends on the random value from the CW and on how long the node has already been waiting for access to the channel. If a new backoff time value is randomized, this waiting time is likely to be extended.

Algorithm 8.4 contains the pseudocode that describes the handling of BE traffic with contention-based phases according to MAC_BH_3a. In the case of MAC_BH_3b, the procedure is the same as in Algorithm 8.3, with the difference that the slot for contention in `MAC_BH_3b` could be larger. The procedures from the pseudocode are clarified in Appendix A. The pseudocode shows how the current backoff time counters are retained and resumed at the beginning of the next phase in `MAC_BH_3a` (Algorithm 8.4 : lines 12 and 15). The pseudocode also shows how `MAC_BH_3b` randomizes a new backoff time value at the beginning of each contention phase (Algorithm 8.3 : lines 11 and 14).

Algorithm 8.4 Fault prevention in the best-effort handling: Pseudocode describing the MAC mechanism `MAC_BH_3a`.

```
 1: procedure MAC_BH_3A
 2:     aifs_prio ← AIFS_AC_VO
 3:     aifs_be ← AIFS_AC_BE
 4:     backoff_time ← 0
 5:     while true do
 6:         wait_for_assigned_slot()
 7:         if is_current_slot_critical() then
 8:             if not is_critical_queue_empty() then
 9:                 m ←dequeue_critical()
10:                 send_csma_ca(m, aifs_prio, 0)
11:             else if not is_be_queue_empty() then
12:                 send_be(aifs_be, backoff_time)
13:         else if is_current_slot_be() and not is_restricted_phase() then
14:             if not is_be_queue_empty() then
15:                 backoff_time ←send_be(aifs_be, backoff_time)
```

In `MAC_BH_3`, the worst-case channel access delay for BE data is unbounded as it depends on the current channel utilization. The best case for BE data is to access directly after a waiting time of $T_{delay} = T_{AIFS}$.

In addition to the mechanisms described for each of the proposals, `MAC_BH_2` and `MAC_BH_3` also require an additional mechanism to prevent a message sent at the end of a BE slot or BE phase from overlapping with the following critical traffic slot and affecting its real-time behavior. For this purpose, a restricted phase is defined at the end of the slot or phase in which new messages cannot start being sent, the channel is not sensed and no backoff time counters are decremented. The duration of the restricted phase corresponds exactly to the time required to complete a transmission, so that a message can be sent shortly before the start of the restricted phase and the transmission finishes at the end of the restricted phase without interfering with the upcoming critical traffic slot. The restricted phase is shown in the pseudocode, specifically in Algorithm 8.3 (line 12) and Algorithm 8.4 (line 13).

The use of DCF and ARQ mechanisms such as RTS/CTS and ACKs, respectively, is not required in the envisaged proposals, but could be introduced if needed by considering slots long enough to accommodate the additional overhead. Also, NAV is only useful for BE transmissions under `MAC_BH_3` and can otherwise be omitted. Table 8.13 summarizes the DCF and ARQ mechanisms that apply to each of the presented BE handling proposals.

Table 8.13: Fault prevention in the best-effort handling: summary of DCF and ARQ mechanisms applicable to the proposals. In the table, "CT" stands for critical traffic.

MAC mechanism	MAC_BH_1		MAC_BH_2		MAC_BH_3	
Traffic type	**CT**	**BE**	**CT**	**BE**	**CT**	**BE**
T_{DIFS}/T_{AIFS} sensing required?	No		Yes		No	Yes
Minimum required T_{slot}	$T_{ST} + T_{data}$		$T_{AIFS} + T_{CW} + T_{data}$		$T_{ST} + T_{data}$	Not enforced
CW required?	No			Yes	No	Yes
Backoff required?	No			Yes	No	Yes
NAV required?	No					Yes
RTS/CTS required?	No					
ACK required?	No					
ARQ required?	No					

8.5.2 Scheduling

Scheduling goal

The described fault-prevention mechanisms for the wireless medium, which handle different types of traffic, require scheduling the utilization of the limited communication resources. The scheduling solution for the fault-prevention in the critical-traffic handling mechanism was introduced in Section 8.4.2. This solution is extended so that:

- Faults are prevented in the best-effort handling. The scheduler performs a simple round-robin allocation in the MAC_BH_1 and MAC_BH_2 mechanisms for BE data in the slots remaining after the allocation of TT/ST and RC/CBS traffic in MAC_CH. MAC_BH_3 does not require any additional scheduling configuration.

Integration into the TTE and TSN scheduling solution to support the best-effort handling

The scheduling algorithm shall first allocate the critical data according to the scheduling constraints described for the fault-prevention in the critical-traffic handling mechanism, as previously explained in Section 8.4.2. Once the critical data is allocated, the next step is to make the slot assignment for BE traffic. The mechanisms for improved

BE handling require straightforward scheduling that does not need to be formulated with FOL constraints or solved along with the other constraints formulated earlier. The remaining time is divided into slots in the mechanisms `MAC_BH_1` and `MAC_BH_2`, and the slots are assigned to the nodes according to the round-robin principle. With `MAC_BH_3`, no allocation of slots to the nodes is required, as a continuous phase is used instead, which is accessed via a contention method.

8.5.3 Simulator

Simulator tool selection and details

The review of similar research works shows several alternatives for simulation tools for computer networks: OPNET, OMNeT++, ns-3 or Matlab. Among these, Matlab could be considered an outsider as it focuses on numerical simulations, which are common in works dealing with physical layer aspects due to the mathematical models that exist to represent physical phenomena. OPNET, OMNeT++ and ns-3 are discrete event simulators.

Discrete event simulators emulate the state of a system at different points in time. The state is influenced by events that trigger a reaction that follows the simulated system logic (Figure 8.8). Such a reaction causes the system to enter a new state where new events can be scheduled. These simulations therefore consist of an initial state that is updated at the time specified by the events. Once all updates to the system state have been completed at a given time, the simulator time is updated to the time at which the next simulation event is scheduled. The simulation is therefore discrete, as time does not evolve continuously, but the time jumps only to the points in time at which events occur. Among the various simulation options, OMNeT++ [124] is selected in this thesis, as it is widely used in related works.

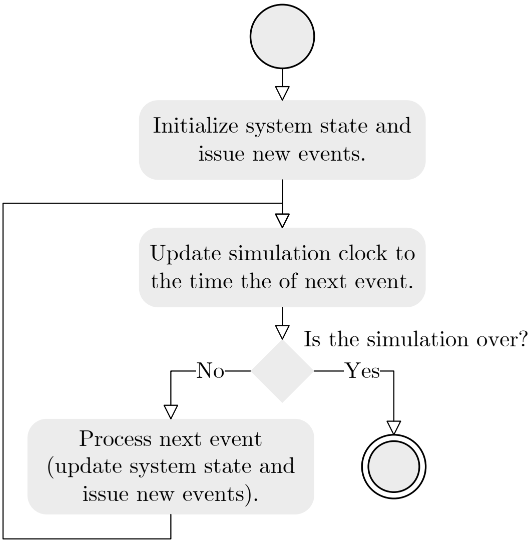

Initialize system state and issue new events.
Update simulation clock to the time the of next event.
Is the simulation over?
No
Yes
Process next event (update system state and issue new events).

Figure 8.8: Workflow of a discrete event simulator

The basic components of any OMNeT++ simulation are the modules, as they are the place where the functionality is implemented. Modules can be connected to each other via gates, and data is exchanged between them via messages. The modules have message handlers that take care of the reaction when a module receives a message. A module is described using the Network Description language (NED) of OMNeT++ in files with the extension `NED`. A NED file includes the module definition by specifying its parameters, gates and nested modules. A module without nested modules is referred to as a simple module and contains the model behavior implemented using the C++ programming language. An initialization file `.ini` is used to assign a value to the parameters declared by the modules. In the initialization file, parameters can be grouped under different labels, each of which provides a different set of values for the same parameters. In this way, it is possible to test different alternatives for the parametrization of the modules and compare their performance, e.g., a module that generates traffic could have a parameter that specifies the amount of traffic generated and takes two different values under two different configuration labels.

The simulation can be executed in a graphical environment or via the command line. To retrieve statistics from the modules, a mechanism is used that is based on signals that are emitted when relevant events occur. A signal can be issued together with any information that can be considered relevant, e.g., when a message is received, a signal containing the length and timestamp of the message can be emitted. The network simulator provides results in three file types: scalar, vector and log. Scalar result files are used to retrieve statistics, e.g., the number of messages sent. Vector result files contain lines with one or more entries each. The recorded data in the vector files can, for example, represent the time at which a particular message was sent. The scalar and vector files can be used for later analysis with internal OMNeT++ tools or read as a text file by external tools, e.g., MATLAB.

Although OMNeT++ provides a framework that allows the creation of modules that resemble the behavior of the different layers in the communication system, it does not provide their implementation. Instead, the INET library [125] provides such modules with models for protocols in the TCP/IP stack, or wired and wireless physical and data-link layers, including Ethernet and IEEE 802.11, among others. The INET modules thus form the basis on which the protocols from the contribution are implemented.

Simulation goals and features

The aim of the developed simulator is to evaluate the performance of the proposed mechanisms based on selected performance metrics:

- **Fault prevention in the best-effort handling.** Performance evaluation of BE data transmissions according to the MAC mechanisms `MAC_BH_1`, `MAC_BH_2a`, `MAC_BH_2b`, `MAC_BH_3a` and `MAC_BH_3b` in terms of delay and reliability for different traffic patterns using the following performance metrics:
  - **Channel access delay**. Reflects the time a message takes from entering the MAC layer to being sent over the transmission medium.

  - **Percentage of collisions**. The percentage of collisions is inversely proportional to the reliability, as collisions are the reason for an unfulfilled service in this simulation.

The simulation developed in this contribution has the following features:

- Simulation of a hybrid network with IEEE 802.11 and Ethernet as technologies for the physical and data-link layers. The baseline MAC protocol for IEEE 802.11 is DCF and the standard version is IEEE 802.11b due to its availability at the time of deploying the simulator, but this does not limit the applicability of the results to newer standard updates. DCF includes DIFS/AIFS, CW, backoff, NAV and RTS/CTS. In addition, ARQ mechanisms, including ACK, are also part of the simulation. The DCF and ARQ mechanisms can be enabled and take different parameters, or be disabled, depending on the simulation configuration. The physical layer simulation of INET calculates a bit-error probability according to the combination of the received signal and the applicable noise, which is often modeled as AWGN. The physical model also considers different path loss models, but by default a free-space path loss model is assumed. The simulator then uses the calculated bit-error probability to indicate whether the reception is successful.
- Modified IEEE 802.11 data-link layer, in particular the MAC layer, to include the following aspects:
  - Dispatching of critical data according to the fault prevention in the critical-traffic handling mechanism `MAC_CH`.
  - Dispatching of BE data according to the fault prevention in the best-effort handling mechanisms `MAC_BH_1`, `MAC_BH_2a`, `MAC_BH_2b`, `MAC_BH_3a` and `MAC_BH_3b`.
  - Implementation for APs and wireless end systems in IEEE 802.11 infrastructure (BSS) and ad-hoc (IBSS) modes.
- New application layer that generates data according to various patterns, including random generation at a given rate with or without being synchronized to the transmission schedule.
- The simulation of clock synchronization is beyond the scope of this contribution, therefore it is assumed that the clocks are perfectly synchronized.

Simulation toolchain

The simulation developed based on OMNeT++ is highly configurable. The simulation requires specifying inputs regarding the network topology, the intended data transmissions, the scheduling configuration or the selected MAC mechanism, among others. Therefore, when different combinations of such inputs are tested in order to evaluate and compare them, considerable effort is required to generate the configuration. This is especially true if the configuration is created manually, not to mention the higher probability of making mistakes.

Therefore, this contribution develops a novel simulation toolchain called Heterogeneous Hybrid Networks Reliability Simulation (HeNReS). HeNReS facilitates the process of generating configurations for simulation, running them and retrieving the performance results by providing a set of tools that perform these tasks, and are straightforward to run. The user interfaces with the tools by providing in a single file a set of simulation parameters that describe the intended simulation configurations. The user then runs HeNReS, which automatically executes all the necessary tools to obtain the performance measures from the simulated configurations.

Internally, HeNReS consists of five steps in which five different applications are executed: network and traffic generator, traffic scheduler, simulation files generator, network simulator and results processing tool. Figure 8.9 shows the workflow of the toolchain, including the executed programs and the representation of their input and output files as interfaces between these programs and the user. Note that Francisco Pozo developed the tools for two of the steps in the context of the RetNet project [5]: the network and traffic generator, and the traffic scheduler.

Step 1. Network and traffic generator. It generates the network topology and the data to be exchanged. The following list summarizes some of the characteristics of the network and traffic generator:

- Models any network topology without cycles, including wired end systems, wireless end systems, switches and APs. The network can be randomly generated by the tool after the user has specified some guidelines, or it can be manually defined by the user.
- Models unicast, multicast and broadcast critical periodic traffic. This traffic can be randomly generated by the tool after the user has specified some guidelines or it can be manually defined by the user.
- Models unicast, multicast and broadcast BE traffic. This traffic shall be specified manually by the user.
- Allows the definition of collision domains that group multiple links. This is useful for modeling wired buses or wireless links.

Step 2. Traffic scheduler. It implements a scheduling algorithm capable of generating the scheduling configuration for periodic critical traffic over large and complex networks. The following list summarizes some of the features of the traffic scheduler:

- Models the scheduling problem for periodic critical traffic using FOL constraints as described in the section of integration into the TTE and TSN scheduling solution to support fault prevention in the critical-traffic handling (Section 8.4.2).
- Attempts to solve the FOL constraints problem using the SMT solver Z3.
- Allocates the BE slots as described in the section of integration into the TTE and TSN scheduling solution to support the best-effort handling (Section 8.5.2).

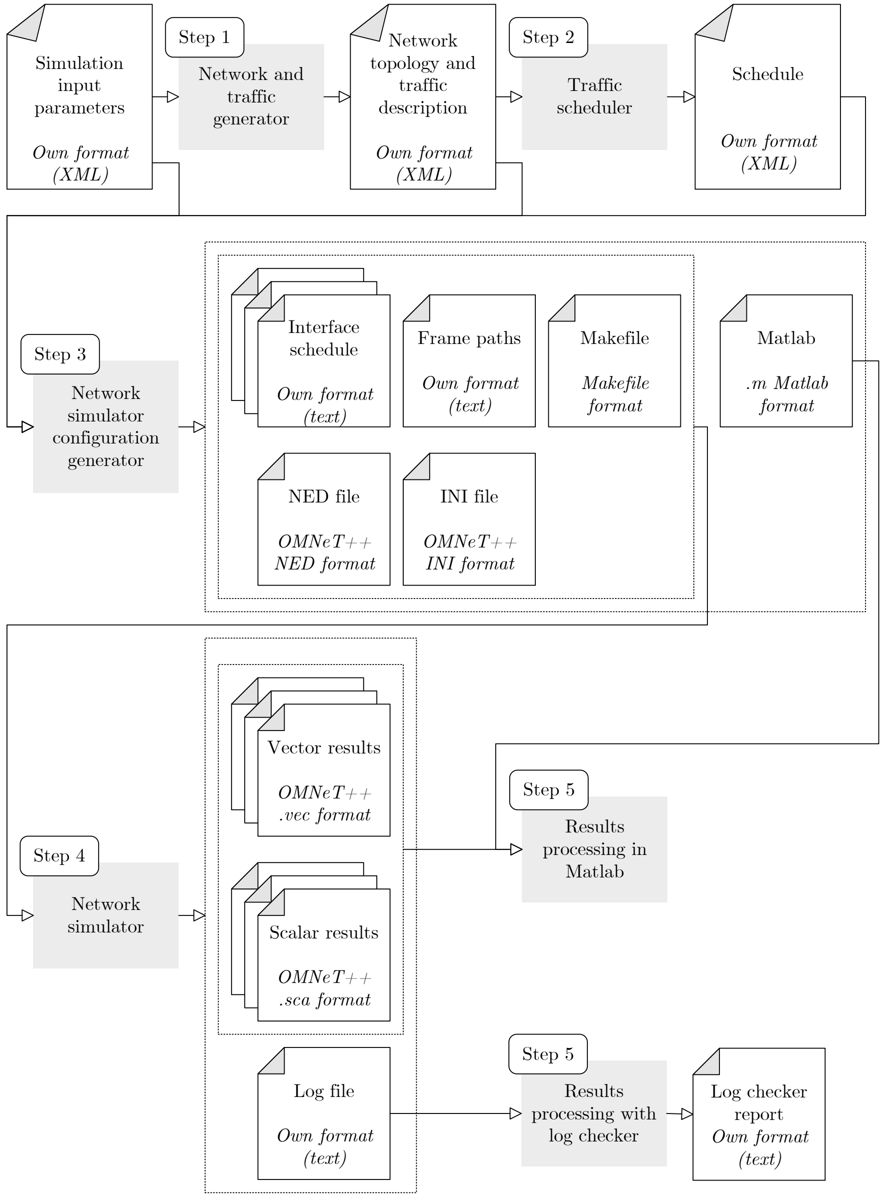


Figure 8.9: Workflow of the simulation toolchain

Step 3. Network simulator configuration generator. This step takes the network topology and traffic specification along with the output of the scheduler and uses them to create and configure the various simulation scenarios in OMNeT++. Thus, this step is responsible for generating the files required by the OMNeT++ simulation modules developed in this contribution. The following list summarizes some of the features of the network simulator configuration generator:

- Generates the NED file, which contains the network topology.
- Generates the INI file, which contains initialization values for the simulation modules, including the MAC mechanisms to be simulated, the characteristics of the traffic to be generated or the characteristics of the induced faults. It also provides different initialization values for the different scenarios to be simulated, grouped under different labels.
- Translates the schedule from Step 2 into a format that is understood by the simulation modules.
- Generates a Makefile that is used to facilitate the execution of the simulation configurations and the processing of the results.
- Generates a Matlab script file to facilitate the import of the simulation data in Matlab for analyzing the results.

Step 4. Network simulator. The actual simulation software developed for OMNeT++ according to the simulation goals and features section.

Step 5. Results processing. The simulation outputs scalar and vector data that are post-processed in Matlab to draw the figures and create the tables that are later shown in the results sections.

In addition, a tool was developed that takes a simulation log file with a list of events as input and generates a report after evaluating whether the logic and timing of the events match the design of the MAC mechanisms and the schedule configuration, i.e., it evaluates to some extent whether the developed simulation is correctly implemented. For this purpose, a log system is added to the simulation that writes an entry in a log file for each relevant event, e.g., time slot, message, channel sensing or restricted phase. For each entry, information is provided about the calling C++ function, the logging MAC address, the time, slot number and the type of action performed by the event, e.g., start of channel sensing and end of channel sensing. A program checks the log file at the end of the simulation and compares its content with the schedule file. Such a comparison is used to report statistics and check general aspects such as timing or the correct allocation of time slots and the correct sequencing of events, e.g., sense the channel before sending in a contention-based approach. In addition, it is also checked whether the recorded channel access delay values are within the limits specified by the theoretical minimum and maximum values. Due to time constraints, the log checker was only developed for the simulations that evaluate fault prevention in the best-effort handling mechanisms. Having such a log checker at an early stage of the research was considered quite relevant, given that it was the first time the OMNeT++ simulation was being developed in this thesis work. Hence, the confidence in the correctness of the developed simulator was not very high, but the log checker tool helped to solve some coding bugs and increase the confidence in the results.

The following list summarizes some of the features of the results processing step:

- Processes the scalar and vector simulation output data in Matlab to create figures and tables to display the results.
- Checks the simulation log for the fault prevention in the best-effort handling mechanisms from `MAC_BH_1`, `MAC_BH_2a`, `MAC_BH_2b`, `MAC_BH_3a` and `MAC_BH_3b`. This is used to evaluate the correctness of the implemented simulator by checking the events logged during the simulation time.

8.5.4 Performance evaluation

The aim of the following experiments campaign is to evaluate through simulations the performance of BE data transmissions according to the MAC mechanisms `MAC_BH_1`, `MAC_BH_2a`, `MAC_BH_2b`, `MAC_BH_3a` and `MAC_BH_3b` in terms of delivery delay and reliability for different traffic patterns.

Scenario description

The setup used to compare different configurations to evaluate their performance comprises a small wireless network with five end nodes and one AP connected according to the topology model (Section 8.3.1). The setup is considered large enough to obtain results that make it possible to identify the differences between the configurations.

As far as the parametrization of the MAC mechanisms is concerned, in the case of `MAC_BH_2` the AIFS for voice applications (AC_VO) is selected for the prioritized node. The AIFS for BE applications (AC_BE) applies to the other nodes. As for the CW, `MAC_BH_2a` does not use any, but in `MAC_BH_2b` the CW size used for video applications (AC_VI) is selected, given that the provided range has enough values to reduce collisions without increasing the maximum delay too much. In `MAC_BH_3`, the default values from AC_BE for AIFS and CW are used. The value for the CW is set to the initial minimum and does not increase. The selected bitrate R is 11 Mbps, the highest possible in IEEE 802.11b. Table 8.14 summarizes the protocol-related parameters of the simulation.

The traffic only goes from the nodes to the AP, i.e., uplink, a common scenario in industrial sensor networks. The size of the exchanged messages $m_i.size$ is relatively small: 62 bytes, of which 4 bytes are the payload. This size was chosen because messages in industrial networks are usually small. In terms of traffic, TT/ST messages are generated periodically, while BE messages are generated randomly, both according to the specified message load. The load is defined as the percentage of occupied slots within each hyper period. To make the results comparable, the application that generates the messages adjusts the rate to the MAC mechanism with the largest slot size, so that for a load of 100%, only the mechanism with the largest slot size is fully loaded. In contrast, all other mechanisms with smaller slot sizes are underutilized, i.e., they do not have a message to send in every slot. A low-medium load of 40% and a high load of 100% are selected for the traffic patterns. These loads can be achieved by combining different types of traffic. The selected combinations include the cases where there is a majority of TT/ST, a majority of BE traffic or a balance between them. In addition, this load can be generated by a single sending node or shared so that all nodes send. Furthermore, all MAC mechanisms are tested under different BE

Table 8.14: Fault prevention in the best-effort handling: protocol-related simulation parameters

Parameter	**Value**
MAC mechanisms	`MAC_BH_1`
	`MAC_BH_2a`
	`MAC_BH_2b`
	`MAC_BH_3a`
	`MAC_BH_3b`
DCF and ARQ characteristics	• $T_{ST} = 20\,\mu$s (IEEE 802.11b) • `MAC_BH_2` (default node): AIFS = 3 slots, $T_{AIFS} = 3T_{ST} = 60\,\mu$s (AC_BE; IEEE 802.11b) • `MAC_BH_2` (prioritized node): AIFS = 2 slots, $T_{AIFS} = 2T_{ST} = 40\,\mu$s (AC_VO; IEEE 802.11b) • `MAC_BH_2a`: CW = 0 slots, $T_{CW} = 0\,\mu$s • `MAC_BH_2b`: CW = [0,15] slots, $T_{CW} = [0,15]T_{ST} = [0,300]\mu s$ (AC_VI; IEEE 802.11b) • `MAC_BH_3`: AIFS = 3 slots, $T_{AIFS} = 3T_{ST} = 60\,\mu$s; CW = [0,15] slots, $T_{CW} = [0,15]T_{ST} = [0,300]\mu s$ (AC_BE; IEEE 802.11b) • Refer to Table 8.13 for additional details.
Wireless data payload rate R (IEEE 802.11b) [Mbps]	11

slot distributions. The BE slot distribution refers to how the BE slots are allocated along the hyper period: either all together after the TT/ST or evenly distributed between TT/ST, i.e., as few consecutive BE slots as possible or as many BE phases as possible, respectively.

When evaluating `MAC_BH_3` for different BE slot distributions, it becomes clear that channel access is not possible if the BE slot distribution is evenly distributed and 50% or more TT/ST are present. The reason why channel access is not feasible is that when the TT/ST percentage is 50% or more, two or more consecutive BE slots are not possible, which in turn means that there is not enough time to complete a transmission in a single slot. Therefore, if it is not possible to influence the scheduler so that BE slots can be scheduled consecutively, `MAC_BH_3` should not be used. Conversely, the BE slot distribution has only a negligible effect on `MAC_BH_1` and `MAC_BH_2`, which affects the minimum delay, but only slightly affects the average delay.

The simulations were run for enough time to have around 500 channel access delay records for each of these combinations. All traffic-related simulation parameters are summarized in Table 8.15.

There is no fault injection in this evaluation, except for the modeled AWGN channel. The free-space model is chosen to simulate the path loss. Faults occur due to

Table 8.15: Fault prevention in the best-effort handling: traffic-related simulation parameters

Parameter	**Value**
Load per traffic class	Low-medium 40%: 20% (TT/ST) - 20% (BE)
	Low-medium 40%: 10% (TT/ST) - 30% (BE)
	Low-medium 40%: 30% (TT/ST) - 10% (BE)
	High 100%: 0% (TT/ST) - 100% (BE)
	High 100%: 20% (TT/ST) - 80% (BE)
	High 100%: 50% (TT/ST) - 50% (BE)
	High 100%: 80% (TT/ST) - 20% (BE)
BE source nodes	One node sending
	All nodes sending
BE slot distribution	Packed together
	Evenly distributed
Data message size $m_i.size$ [bytes]	62
TT/ST data message generation pattern	Periodically and in bursts of as many messages as the indicated load allows.
BE data message generation pattern	Randomly and as many messages as the indicated load allows.
Number of transmitted BE data messages $\|M\|$	As many as to have ≥ 500 channel access delay records for each configuration.

collisions between the generated BE traffic. Consequently, TT/ST is expected to suffer no faults since the schedule is created in such a way that no collisions occur.

The combination of various protocol-related configurations (MAC mechanisms) and traffic-related parameters (load per traffic class, BE source nodes and BE slot distribution) results in a total of 140 different configurations/scenarios to compare.

Results

Figure 8.10 shows the average channel access delay for `MAC_BH_1`. It can be observed that this MAC mechanism is not good when there is only one sender, but performs very well when all nodes are sending. The good performance in the latter case is a clear benefit of the round-robin mechanism, which evenly distributes the opportunities to access the medium. However, suppose there is only one sender. In this case, all channel access opportunities assigned to the non-sender nodes are lost, and the channel access delay for the sender node increases significantly. It can also be appreciated that the delay increases with the traffic load and the BE traffic is particularly affected by the amount of TT/ST. However, the main benefit of `MAC_BH_1` is that there are no collisions and thus there is a predictable, upper-bounded channel access delay for both TT/ST and BE traffic.

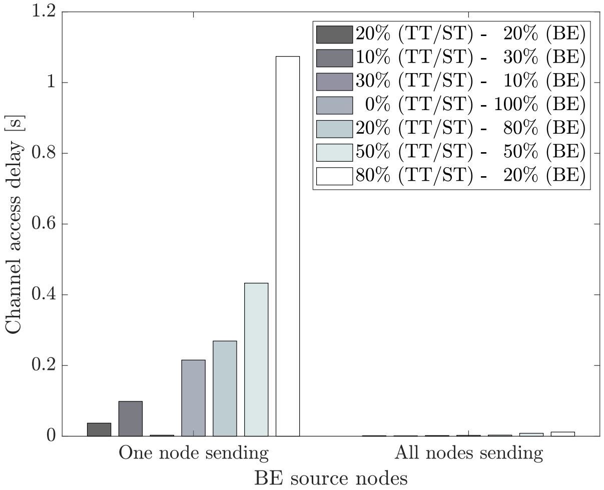


Figure 8.10: Fault prevention in the best-effort handling: average channel access delay for MAC_BH_1 under different loads per traffic class and BE traffic source nodes options

Figure 8.11 shows the average channel access delay for MAC_BH_2. If no contention is allowed (MAC_BH_2a), the behavior is similar to MAC_BH_1. However, if contention is allowed (MAC_BH_2b), the difference in channel access delay between the traffic sent by one or all nodes is negligible. Unfortunately, Figure 8.12 shows that collisions occur when data originates from multiple nodes, and that the number of collisions increases when the CW is allowed.

Figure 8.13 shows the results for the average channel access delay with MAC_BH_3. The results show that the protocol option of storing the backoff time counter in each phase (MAC_BH_3a) or restarting it (MAC_BH_3b) has no significant effect and is only slightly lower in the case of storing the backoff time value between phases. Further, the channel access delay is much lower than for MAC_BH_1 and MAC_BH_2. However, the Figure 8.14 clarifies that the price to pay is that the number of collisions is relatively high when all nodes are sending.

Figure 8.15 summarizes the mean values for the channel access delay for all the different MAC mechanisms. For MAC_BH_2 and MAC_BH_3 only the best configuration options of the mechanism are shown, i.e. MAC_BH_2b and MAC_BH_3b, respectively. Given these results, it can be appreciated that MAC_BH_3 yields the lowest average channel access delay unless 50% or more of the traffic is reserved for TT/ST and the time slots are evenly distributed. MAC_BH_1 is best when all nodes are sending traffic, both in terms of average and guaranteed maximum channel access delay. As for the number of collisions, MAC_BH_2 and MAC_BH_3 have no collisions when there is only one sender. Since they have a lower channel access delay than MAC_BH_1, they are the preferred options when traffic emerges from only one node. When the load distribution is unknown, MAC_BH_2b provides the best compromise as the worst-case delay is bounded and collisions only occur when the prioritized node has nothing to send.

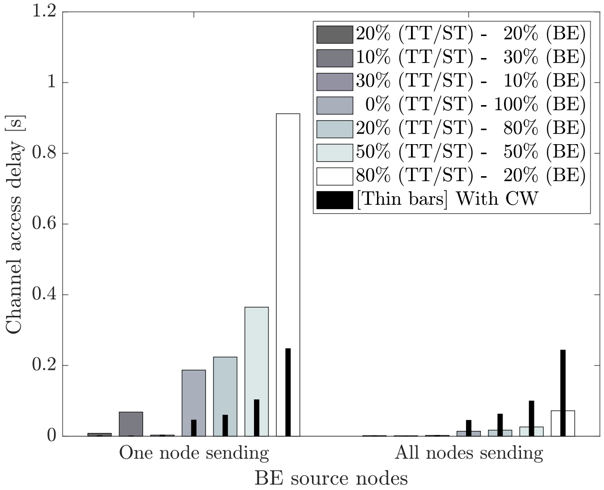


Figure 8.11: Fault prevention in the best-effort handling: average channel access delay for MAC_BH_2a and MAC_BH_2b under different loads per traffic class and BE traffic source nodes options. The wide bars correspond to MAC_BH_2a (without CW), whereas the thin black bars represent the same underlying configuration but for MAC_BH_2b (with CW).

8.5.5 Conclusions

This contribution presents fault-prevention mechanisms for BE traffic MAC_BH_1, MAC_BH_2 and MAC_BH_3 to overcome the challenges of using a half-duplex medium when transmitting over IEEE 802.11, which might cause interference from other users. Specifically, MAC_BH_1 proposes pre-scheduled BE time slots with a round-robin assignment, MAC_BH_2 proposes contention-based BE time slots with round-robin assignment for a prioritized node, and MAC_BH_3 proposes contention-based BE phases. The three mechanisms are integrated into the scheduling solution introduced for MAC_CH. A simulation toolchain named HeNReS is developed to simulate the proposed mechanisms. The simulation results clearly show that the choice of each of the three proposed mechanisms depends on the traffic pattern. The more is known about the traffic, the better is the performance that can be achieved by selecting the appropriate mechanism. In particular, when the data originates from one node, MAC_BH_2 and MAC_BH_3 are preferable, while MAC_BH_1 is most suitable when the data is evenly distributed among the nodes. Conversely, if nothing is known, MAC_BH_2 is the best compromise.

MAC_BH_1, MAC_BH_2 and MAC_BH_3 not only provide higher reliability and better timing performance, but also provide better availability for BE data. The mechanisms based on pre-assigned slots help to reduce uncertainties and support determinism. The addition of these mechanisms leads to more complexity and costs, but these are not excessive due to their simplicity. The mechanisms do not lead to additional overhead penalties. Finally, MAC_BH_1 does not exhibit adaptability as it is based on a fixed schedule. In contrast, MAC_BH_2 and MAC_BH_3 respond better to dynamic changes.

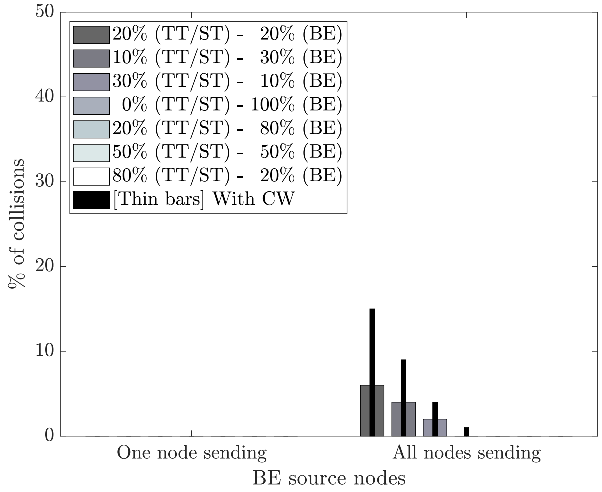


Figure 8.12: Fault prevention in the best-effort handling: percentage of collisions for MAC_BH_2a and MAC_BH_2b under different loads per traffic class and BE traffic source nodes options. The wide bars correspond to MAC_BH_2a (without CW), whereas the thin black bars represent the same underlying configuration but for MAC_BH_2b (with CW).

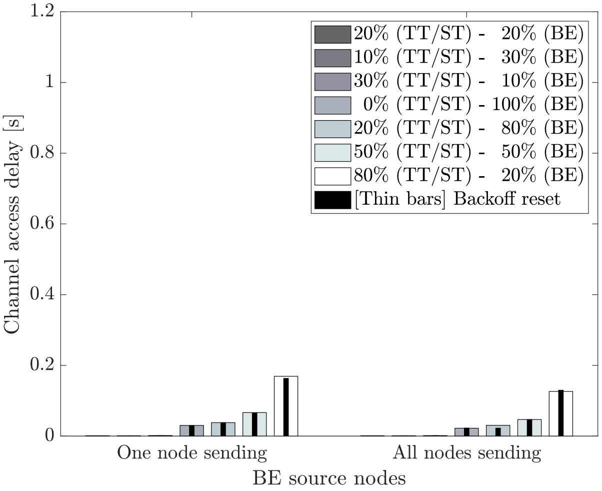


Figure 8.13: Fault prevention in the best-effort handling: average channel access delay for MAC_BH_3a and MAC_BH_3b under different loads per traffic class and BE traffic source nodes options. The wide bars correspond to MAC_BH_3a (without backoff time counter reset), whereas the thin black bars represent the same underlying configuration but for MAC_BH_3b (with backoff time counter reset).

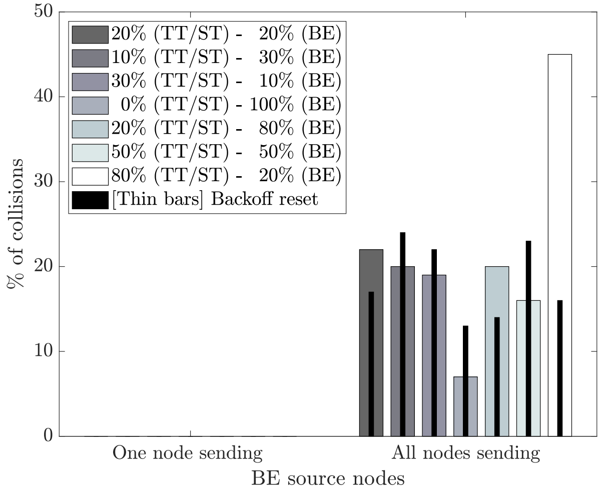


Figure 8.14: Fault prevention in the best-effort handling: percentage of collisions for `MAC_BH_3a` and `MAC_BH_3b` under different loads per traffic class and BE traffic source nodes options. The wide bars correspond to `MAC_BH_3a` (without resetting the backoff time counter), whereas the thin black bars represent the same underlying configuration but for `MAC_BH_3b` (resetting the backoff time counter).

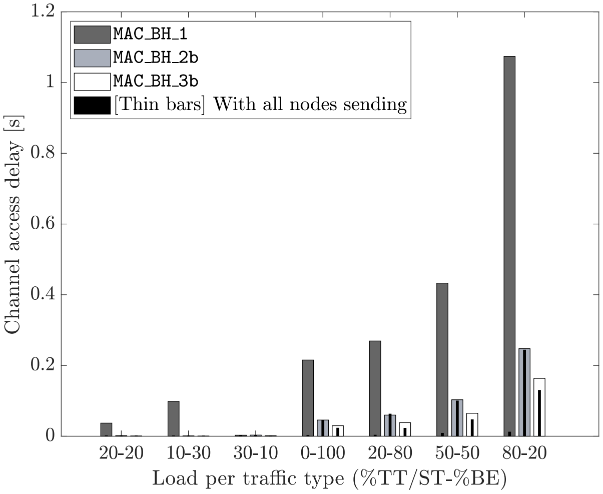


Figure 8.15: Fault prevention in the best-effort handling: average channel access delay for `MAC_BH_1`, `MAC_BH_2` and `MAC_BH_3` under different loads per traffic class. For `MAC-_BH_2` and `MAC_BH_3` only the best configuration options of the mechanisms are shown, i.e., `MAC_BH_2b` and `MAC_BH_3b`, respectively. The wide bars correspond to the case where all BE traffic is sent by one node, whereas the thin black bars represent the same underlying configuration but for the case where all nodes send BE traffic.

8.6 Fault prevention using cognitive radio (MAC_CR)

8.6.1 Mechanism description

Applying frequency diversity in the form of cognitive radio [5] is a fault-prevention mechanism that is expected to significantly reduce the probability of faults occurring, especially in environments full of interference (as initially indicated in Table 7.9 in the fault-handling proposal). Since cognitive radio selects the transmission frequency that minimizes the probability of suffering interference, the mechanism applies to all transmissions, without differentiation by traffic class. The mechanism is labeled as MAC_CR.

To support cognitive radio capabilities in the wireless MAC protocol, cognitive radio phases are periodically allocated within the wireless MAC protocol. During these phases, no data transmissions can take place. The approach is similar to [117], a proposal evaluated in [119] in different industrial environments. Figure 8.16 shows an example of the time allocation for the proposed cognitive radio scheme. The generated schedules, which take into account the handling of all traffic classes, are distributed to the participant nodes before the network is put into operation. More details about the scheduling solution to handle cognitive radio phases are provided later in Section 8.6.2.

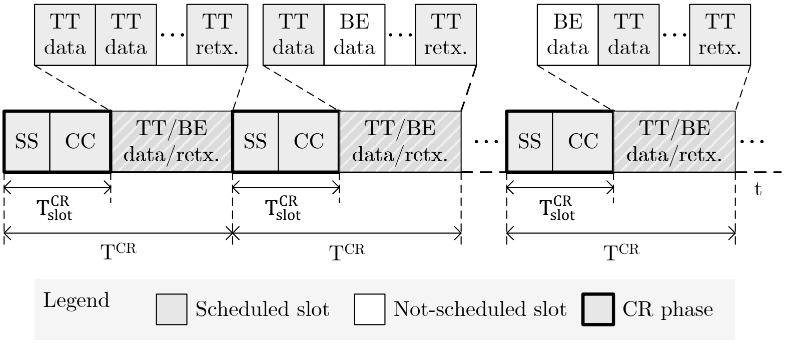


Figure 8.16: Fault prevention using cognitive radio: example of time allocation

Cognitive radio phases are characterized by the duration T_{slot}^{CR} and the period at which they occur T^{CR}. These periodic phases are divided into a spectrum sensing (SS) phase and a cognitive radio control (CC) phase. During the SS phases, the devices perform SS tasks to collect environmental information. In this case, the SS algorithm enables the devices to detect whether a channel is free or being interfered with by an external user. Since the goal in each AP coordinating the cognitive radio procedure is to find a common channel that is least compromised by external interference, all devices in the network perform the SS task simultaneously. Each device senses a different channel assigned to it by the scheduler during SS. In this way, as many channels as there are devices in the network can be sensed during a cognitive phase. Moreover, the channel sensed by a device changes from one cognitive phase to the next, so that the network can maximize the spatial diversity of all devices. As soon as the sensing results are available, all devices send them to the AP in the CC phase.

The AP uses the SS results to obtain statistics on the external interference of all channels. First, a Markov process [126] estimates the availability probability of the individual channels. Then the AP creates a common list of channels ordered by their availability, which is sent to all devices in the AP coverage area. Each device then synchronizes its radio to the first channel in the shared list. In this scheme, the system selects the least interfered channel. However, external interference can disrupt the channel through which the network exchanges information during the cognitive phase, making the exchange of cognitive radio data impossible. To solve this problem, all devices in the network check whether the other devices have sent their results during the cognitive radio phase, and if no results are received from other devices, the channel is considered interfered. In this case, the devices use the list sent in the previous cognitive phase to switch to the next channel in the list and continue their operation.

Algorithm 8.5 contains the pseudocode that describes the cognitive radio mechanism from the perspective of the end systems and APs. The procedures from the pseudocode are clarified in Appendix A. Some of the most relevant aspects that the pseudocode shows are: The SS phase (lines 6 and 23), the CC phase (lines 9 and 26), how each device senses a different channel (lines 7 and 24), how the sensed channel changes from one cognitive phase to the next (lines 8 and 25), and how the sensing results are sent to the AP in the CC phase (line 10).

Other aspects shown by the pseudocode are: How the SS results are received by the AP (line 27), how the AP creates the list of channels, ordered by availability (line 28) and how the list is then sent to all devices in the AP coverage area (lines 11 and 29), the synchronization of each device's radio with the first channel in the list (lines 15-17, 33-35), checking whether other devices have sent their results during the cognitive radio phase (lines 12 and 30), and how the devices switch to the next channel in the list if the current channel is interfered (lines 13, 17, 31 and 35).

8.6.2 Scheduling

Scheduling goal

The described fault-prevention mechanisms for the wireless medium, which handle different types of traffic, require scheduling the utilization of the limited communication resources. The scheduling solution for the fault-prevention in the critical-traffic handling mechanism was introduced in Section 8.4.2. This solution is extended to handle the resources shared by multiple nodes sending and receiving data such that:

- Faults are prevented by the use of cognitive radio, enabling the `MAC_CR` mechanism. TT/ST, RC/CBS and BE traffic are transmitted over the wireless medium with a higher delivery probability. For this purpose, periodic phases must be scheduled in which no data transmissions take place but the cognitive radio algorithm is running.

Algorithm 8.5 Fault prevention using cognitive radio: Pseudocode describing the MAC mechanism MAC_CR from the point of view of end systems (mac_cr_es) and APs (mac_cr_ap).

```
procedure MAC_CR_ES_i
    curr_ss_channel ← i
    curr_tx_channel ← 0
    avbl_channels ← ALL_CHANNELS
    while true do
        wait_for_ss()
        channel_info ←check_channel(curr_ss_channel)
        curr_ss_channel ←(curr_ss_channel + 1) mod length(ALL_CHANNELS)
        wait_for_cc()
        send_channel_info_to_ap(channel_info)
        new_avbl_channels ← recv_avbl_channels_from_ap()
        if is_avbl_channels_corrupt(new_avbl_channels) then
            curr_tx_channel ←(curr_tx_channel + 1) mod length(ALL_CHANNELS)

        else
            avbl_channels ← new_avbl_channels
            curr_tx_channel ← 0
        switch_tx_channel(avbl_channels[curr_tx_channel])

procedure MAC_CR_AP_i
    curr_ss_channel ← i
    curr_tx_channel ← 0
    avbl_channels ← ALL_CHANNELS
    while true do
        wait_for_ss()
        channel_info ←check_channel(curr_ss_channel)
        curr_ss_channel ←(curr_ss_channel + 1) mod length(ALL_CHANNELS)
        wait_for_cc()
        channels_info ← recv_channel_info_from_ess()
        new_avbl_channels ← create_avbl_channels_list(channels_info)
        send_avbl_channels_to_ess(new_avbl_channels)
        if is_avbl_channels_corrupt(new_avbl_channels) then
            curr_tx_channel ←(curr_tx_channel + 1) mod length(ALL_CHANNELS)

        else
            avbl_channels ← new_avbl_channels
            curr_tx_channel ← 0
        switch_tx_channel(avbl_channels[curr_tx_channel])
```

Integration into the TTE and TSN scheduling solution to support fault-prevention using cognitive radio

Section 8.4.2 previously described the scheduling constraints for the fault-prevention in the critical-traffic handling mechanism. One of these constraints, Equation 8.3, can also be used to consider the cognitive radio phases. These phases can be modeled as contention-free constraints since no traffic can be sent over the wireless links during the cognitive radio phases. The cognitive radio phases are characterized by the duration T^{CR}_{slot} and the period at which they occur T^{CR}, so that they can be assigned time as if they were critical traffic.

8.6.3 Simulator

Simulator tool selection and details

Computer simulations based on OMNeT++ are used to evaluate `MAC_CR`. The simulator first introduced for the evaluation of the best-effort handling mechanisms is used and extended in this contribution. A description of why OMNeT++ was chosen as the simulation tool and details on the simulation development can be found in Section 8.5.3.

Simulation goals and features

The aim of the developed simulator is to evaluate the performance of the proposed mechanisms based on selected performance metrics:

- **Fault prevention using cognitive radio.** Performance evaluation of data transmissions after using the cognitive radio mechanism `MAC_CR` in combination with the fault-tolerance mechanisms `MAC_TD_Cons_1` and `MAC_TD_Sprd_1` compared to the case where none of these mechanisms are present (`MAC_CR_None`[2] and `MAC_TD_None`[3]). The mechanisms `MAC_TD_Cons_1` and `MAC_TD_Sprd_1` are based on retransmissions and belong to Contribution area 3. Further details on them are described later in Section 9.3. The evaluation is done in terms of delays and reliability for different interference patterns using the following performance metrics:

  – **Average delay from MAC layer to MAC layer**. The delay from MAC layer to MAC layer is the time a message takes from entering the MAC layer on the sender side to leaving the MAC layer on the receiver side. This takes into account the potential retransmissions that the message might experience until it is finally received. This performance metric compares the average delay caused by the retransmission schemes, which increases with each failed transmission attempt.

  – **Percentage of failed messages**. It is inversely proportional to the unfulfilled service. Since the scheduler guarantees that no retransmissions will take place after the message deadline has expired, a deadline overrun is taken into account each time a message is not successfully delivered after the last scheduled retransmission.

[2] In the performance evaluation, the label `MAC_CR_None` is used to refer to the scenarios where `MAC_CR` is not used.

[3] In the performance evaluation, the label `MAC_TD_None` is used to refer to the scenarios where `MAC_TD_Cons_1` and `MAC_TD_Sprd_1` are not used.

The simulator first introduced for the best-effort handling mechanisms (Section 8.5.3) is extended by the following features:

- Modified IEEE 802.11 data-link layer, in particular the MAC layer, to include the following aspects:
  - Added fault prevention using the cognitive radio protocol MAC_CR to select a transmission frequency for each traffic class that maximizes the probability of delivery. The implementation is a joint work with Pedro Manuel Rodríguez [5].
- Fault-injection mechanisms. Modified IEEE 802.11-based nodes to be capable of generating persistent/jamming interference. The default INET modules can already generate CSMA/CA interference. In the cognitive radio based MAC, the interference follows a FH sequence over five different channels.

Simulation toolchain

The simulation toolchain HeNRES, first introduced for the best-effort handling mechanisms (Section 8.5.3), is extended to include the simulation of MAC_CR. The changes for each of the steps in the toolchain are explained below:

Step 1. Network and traffic generator. No modifications in this contribution.

Step 2. Traffic scheduler. The following features are added in this contribution:

- Models the scheduling problem for periodic critical traffic using FOL constraints as previously described in the scheduling section.

Step 3. Network simulator configuration generator. No modifications in this contribution.

Step 4. Network simulator. The network simulator is extended to include the aspects previously mentioned in the simulation goals and features section.

Step 5. Results processing. No modifications in this contribution. Further, the log checker tool is not extended and does not cover MAC_CR.

8.6.4 Performance evaluation

The aim of the following experiments campaign is to evaluate through simulations the performance of data transmissions after using the cognitive radio MAC mechanism MAC_CR. The mechanism is also combined in the experiments with the fault-tolerance mechanisms based on time diversity MAC_TD_Cons_1 and MAC_TD_Sprd_1, which are part of Contribution area 3 and will be presented later in Section 9.3. The performance of these MAC mechanisms is compared to the case when there are no mechanisms, i.e., no cognitive radio (MAC_CR_None) and no time diversity (MAC_TD_None). The evaluation is carried out in terms of delays and reliability for different interference patterns.

Scenario description

A wireless network consisting of five wireless end systems and one AP connected according to the topology model (Section 8.3.1) is used in the simulator to evaluate the proposed MAC mechanisms. The setup is considered large enough to obtain results that make it possible to identify the differences between the configurations.

The bitrate R is set to 11 Mbps, the highest rate offered by IEEE 802.11b. The simulation model allows the choice between time diversity or cognitive radio to increase reliability, and both or neither of the mechanisms may be used simultaneously. The different combinations of MAC mechanisms result in six configurations that are simulated. For time diversity, the maximum number of allowed transmitted copies $m_i^{[v_j,v_k]}.n$ for each message m_i on each dataflow link $[v_j, v_k]$ is set to 2 (1 transmission and 1 retransmission), as this shortens the simulation time while still showing the necessary performance differences. In the cognitive radio based MAC case, the system hops among five different channels according to the spectrum sensing results. The same happens for the FH-based interference. The protocol-related parameters are summarized in Table 8.16.

TT/ST messages with a size $m_i.size$ of 62 bytes are generated in the end systems and sent to other end systems via the AP. In each simulation, at least 5000 messages per configuration are transmitted. The messages are randomly generated with enough messages to fill the scheduled slots in the configuration able to cope with the least amount of traffic. The traffic-related simulation parameters are summarized in Table 8.17.

To evaluate the proposed MAC mechanisms, an AWGN channel is considered. The free-space model is chosen to simulate the path loss. The network is installed in an environment with interference, and two types of interference are considered. Intelligent interference, which does not attempt to interrupt an ongoing transmission, such as CSMA to represent IEEE 802.11 or IEEE 802.15.4 users, and jamming interference, as caused by electromagnetic interference due to industrial equipment. Two different interference levels are simulated (10% and 30% of the total time), transmitted in bursts of different sizes (1, 15 or 30 times $m_i.size$). The scheduler uses the latter to set the value of T_{ITI}, one of the parameters used to configure the time diversity mechanisms. Furthermore, both FH-based interference and static interference are considered. The former models, e.g., Bluetooth systems, while IEEE 802.11 transmissions use the latter. The worst-case interference scenario is represented by a jamming interference taking 30% of the time, an extreme situation, i.e., not many protocols can cope with this high level of interference. Table 8.18 shows a summary of the fault-injection-related parameters.

The combination of various protocol-related parameters (MAC mechanisms) and fault-injection parameters (types of interference, level of interference, interference burst size and FH) results in a total of 144 different configurations/scenarios to compare.

Table 8.16: Fault prevention using cognitive radio: protocol-related simulation parameters.

Parameter	**Value**
MAC mechanisms	Setup 1: MAC_CR_None & MAC_TD_None
	Setup 2: MAC_CR_None & MAC_TD_Cons
	Setup 3: MAC_CR_None & MAC_TD_Sprd
	Setup 4: MAC_CR & MAC_TD_None
	Setup 5: MAC_CR & MAC_TD_Cons
	Setup 6: MAC_CR & MAC_TD_Sprd
DCF and ARQ characteristics	• $T_{ST} = 20\,\mu$s (IEEE 802.11b) • $T_{SIFS} = 10\,\mu$s (IEEE 802.11b) • $T_{DIFS} = T_{SIFS} + 2T_{ST} = 50\,\mu$s (IEEE 802.11b) • CW = [0,31] slots, $T_{CW} = [0,31]T_{ST} = [0,620]\mu s$ (IEEE 802.11b) • $T_{ack} = 203\,\mu$s • Refer to Table 9.4 for additional details.
Wireless data payload rate R (IEEE 802.11b) [Mbps]	11
Maximum number of transmitted copies $m_i^{[v_j,v_k]}.n$	2
Cognitive radio phase duration T_{slot}^{CR} [ms]	2.4
Cognitive radio phase period T^{CR} [ms]	240
Number of FH channels for cognitive radio	5

Table 8.17: Fault prevention using cognitive radio: traffic-related simulation parameters.

Parameter	**Value**
TT/ST data message size $m_i.size$ [bytes]	62
TT/ST data message generation pattern	Randomly and as many messages as the generated schedule coping with the least amount of traffic allows.
Number of transmitted TT/ST data messages $\|M\|$	$\geq$ 5000 for each configuration

Results

Figure 8.17 shows the percentage of failed messages for the case of MAC_CR_None and MAC_TD_None (Setup 1), and MAC_CR combined with MAC_TD_None (Setup 4). The

Table 8.18: Fault prevention using cognitive radio: fault-injection-related simulation parameters.

Parameter	**Value**
Channel model	AWGN Free-space path loss model
Type of interference	CSMA
	Jamming
Level of interference	10%
	30%
Interference burst size (as multiple of $m_i.size$)	1x
	15x
	30x
Frequency hopping (FH)	Yes
	No

results are given for the interference types CSMA and jamming, each under different loads and burst sizes. In Setup 1, the results are shown for both static interference and FH-based interference. However, in Setup 4, no results are shown for static interference as the failed messages are 0, given that the system hops to a non-jammed channel as soon as interference is detected. From these results it can be seen that jamming interference leads to a higher number of failed messages than CSMA, as CSMA devices back off when the medium is busy. On the other hand, the results also show a lower number of failed messages for MAC_CR in Setup 4. Therefore, the proposed cognitive radio scheme increases the robustness of the system in interfered environments by reducing the number of errors. The number of failed transmissions is reduced from more than 50% for jamming interference or 30% in the CSMA case to less than 10% for both. In this comparison, the worst cases correspond to static interference in MAC_CR_None (Setup 1) and FH-based interference in MAC_CR (Setup 4). Regarding the load and the burst size of the interference, MAC_CR_None and MAC_CR show different behavior. For MAC_CR_None (Setup 1), the lower the burst size, the higher the error rate. This is because a smaller burst size also means a more frequent interval for the same level of interference. On the other hand, in the case of MAC_CR (Setup 4), the lowest error rate is achieved with a larger burst size. The reason for this is that a less frequent interference interval makes MAC_CR assume that the channel is free, which results in a higher collision probability than with more frequent interference intervals.

The percentage of failed messages with the schemes MAC_TD_Cons (Setup 2 and Setup 5) and MAC_TD_Sprd (Setup 3 and Setup 6) is shown in Figure 8.18 and Figure 8.19, respectively. From the comparison of the two figures, it can be seen that MAC_TD_Cons is a better policy for CSMA interference, while MAC_TD_Sprd is better for jamming interference, as concluded in the results of the time diversity mechanisms (Section 9.3.5). The two time diversity policies can handle CSMA interference as the

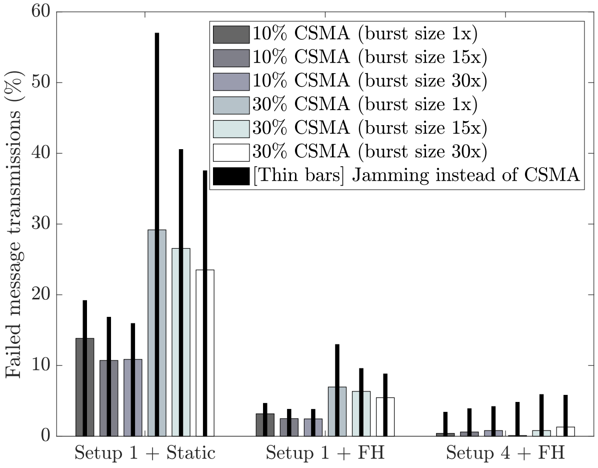


Figure 8.17: Fault prevention using cognitive radio: Percentage of failed messages for MAC_CR_None combined with MAC_TD_None (Setup 1) under static and FH interference, and MAC_CR combined with MAC_TD_None (Setup 4) under FH interference. The wide bars correspond to CSMA interference, whereas the thin black bars represent the same underlying configuration but for jamming interference.

error rate is reduced to 0.05% when using MAC_TD_Cons (Setup 2 and Setup 5) and to 3.5% when using MAC_TD_Sprd (Setup 3 and Setup 6). Unfortunately, the jamming interference is still quite harmful for both. In this case, the approach of MAC_CR is useful because it achieves better results with MAC_TD_None, as shown in Figure 8.17, than MAC_TD_Cons or MAC_TD_Sprd alone. Moreover, even better results are obtained when both cognitive radio and time diversity are combined, as shown for the case of using MAC_CR combined with MAC_TD_Cons (Setup 5) in Figure 8.18, and for the use of MAC_CR in combination with MAC_TD_Sprd (Setup 6) in Figure 8.19. In these two setups, the CSMA interference errors are reduced to 0%, while in Setup 6 only 4% of transmissions fail in the event of jamming interference.

The average MAC-to-MAC delay is measured for all the setups described above. Figure 8.20 and Figure 8.21 show the delay with MAC_TD_Cons (Setup 2 and Setup 5) and MAC_TD_Sprd (Setup 3 and Setup 6), respectively. When using MAC_TD_None (Setup 1 and Setup 4), the delay is always the same for all MAC mechanisms considered, as each message is only transmitted once and discarded if it fails. Hence, the results for MAC_TD_None are not shown. The average delays are shown for different loads and burst lengths of CSMA and jamming interference. In addition, the minimum and maximum delays are shown in dashed lines in both figures. The minimum and maximum delays correspond to the cases in which the message is not retransmitted or when it is retransmitted once, respectively. The results for MAC_CR_None (Setup 1-3) are shown for static interference and FH-based interference, whereas the results for MAC_CR are only shown for FH-based interference. The reason for this is the same as before: MAC_CR hops to a free channel and the results are the same as if there is no

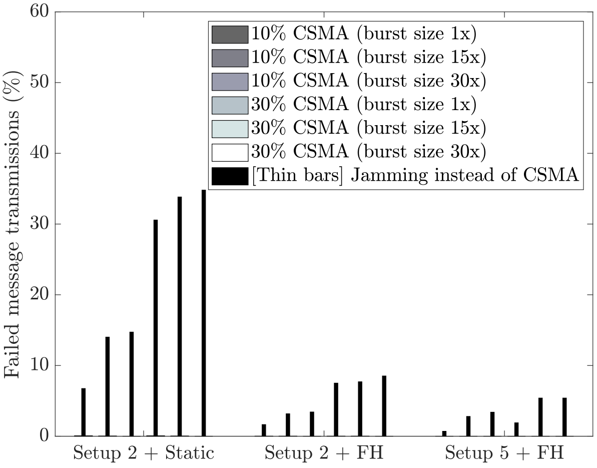


Figure 8.18: Fault prevention using cognitive radio: Percentage of failed messages for `MAC_CR_None` combined with `MAC_TD_Cons` (Setup 2) under static and FH interference, and `MAC_CR` combined with `MAC_TD_Cons` (Setup 5) under FH interference. The wide bars correspond to CSMA interference, whereas the thin black bars represent the same underlying configuration but for jamming interference.

interference. Under `MAC_TD_Cons` (Setup 2 and Setup 5), the delay does not increase significantly in any of the cases considered. With `MAC_TD_Sprd` (Setup 3 and Setup 6), however, the delay and especially the jitter of the delay is much greater, as the additional transmission slots are more spaced apart. For `MAC_CR` (Setup 5-6), the delay is lower as a smaller number of retransmissions are required due to its ability to avoid interference.

As for jitter, it only occurs when the time diversity schemes are involved. Jitter of 240 μs and 4.9 ms are obtained for `MAC_TD_Cons` (Setup 2 and Setup 5) and `MAC_TD_Sprd` (Setup 3 and Setup 6), respectively.

Considering both the failed messages and the average delay, `MAC_CR` in combination with `MAC_TD_None` (Setup 4) achieves fewer failures than time diversity (`MAC_TD_Cons` or `MAC_TD_Sprd`) with `MAC_CR_None` (Setup 2 and Setup 3), and it does not lead to an increase in delay or jitter, which is the desired goal for real-time communications. If the applications require a higher reduction in failed messages, `MAC_CR` might be used together with `MAC_TD_Cons` (Setup 5) or `MAC_TD_Sprd` (Setup 6), but at the cost of an increase in average delay.

8.6.5 Conclusions

This contribution proposes a fault-prevention technique based on cognitive radio, `MAC_CR`. To add cognitive radio, periodic phases are scheduled that allow the protocol to collect data about the transmission channels and make a decision about which channel to use for transmission based on the one that is expected to be more reliable.

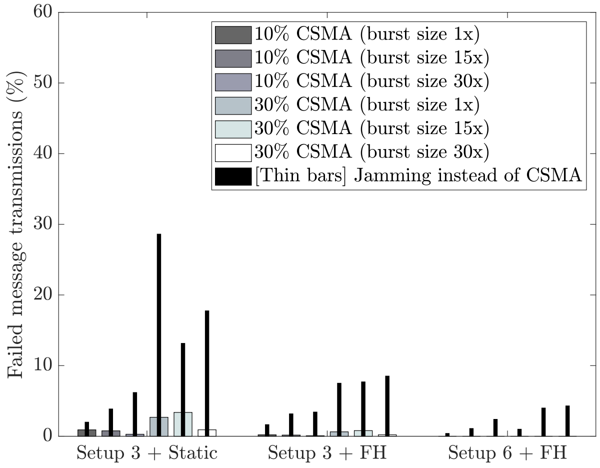


Figure 8.19: Fault prevention using cognitive radio: Percentage of failed messages for MAC_CR_None combined with MAC_TD_Sprd (Setup 3) under static and FH interference, and MAC_CR combined with MAC_TD_Sprd (Setup 6) under FH interference. The wide bars correspond to CSMA interference, whereas the thin black bars represent the same underlying configuration but for jamming interference.

The cognitive radio mechanism is integrated into the scheduling solution introduced for MAC_CH. The performance of the protocol is compared with the fault-tolerance mechanisms MAC_TD_Cons and MAC_TD_Sprd that are proposed in Contribution area 3, which are based on retransmissions. The evaluation is performed in environments that have been exposed to interference of different types, duration and persistence. The results of the simulation in HeNReS show that the MAC_CR scheme achieves better results than MAC_TD_Cons and MAC_TD_Sprd in all environments considered, as the cognitive radio scheme can reduce the percentage of failed messages more than the time diversity schemes. The percentage of failed messages is reduced to less than 10% when using MAC_CR, while percentages of more than 30% are achieved when using MAC_TD_Cons and MAC_TD_Sprd for some interference patterns. Moreover, this reduction does not come at the expense of an increase in the average MAC-to-MAC delay, as in the case of MAC_TD_Cons and MAC_TD_Sprd. On the other hand, MAC_CR leads to some disadvantages, such as lower energy efficiency and also consumes part of the throughput. In addition, a scheme that uses both time diversity and cognitive radio is evaluated and achieves an even greater reduction in the number of failed messages. The percentage of failed messages is below 4% when both techniques are used simultaneously. However, in this case, an increase in the average MAC-to-MAC delay must be taken into account as the scheme includes retransmissions.

The main goal of MAC_CR is to improve reliability and enable the support for highly critical data exchanges. However, improvements in availability and safety are also expected. Furthermore, the use of cognitive radio is expected to favor the adaptability to dynamic changes in channel conditions. Determinism depends on the conditions for

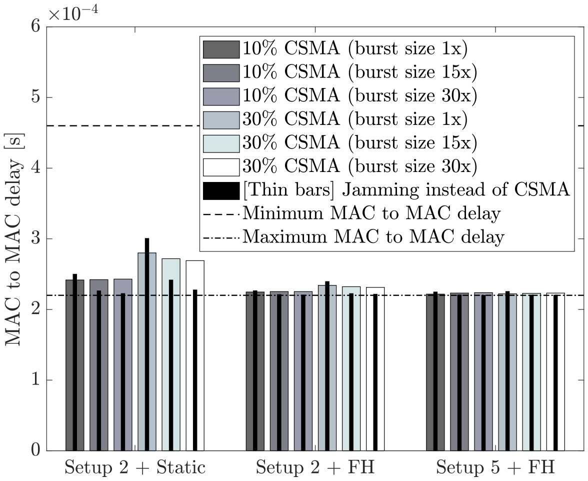


Figure 8.20: Fault prevention using cognitive radio: Average MAC-to-MAC delay for `MAC_CR_None` combined with `MAC_TD_Cons` (Setup 2) under static and FH interference, and `MAC_CR` combined with `MAC_TD_Cons` (Setup 5) under FH interference. The wide bars correspond to CSMA interference, whereas the thin black bars represent the same underlying configuration but for jamming interference.

which it is defined, but should benefit from a higher proportion of successful transmissions. However, these benefits do not come for free. Throughput decreases due to protocol overhead, as the system cannot transmit data during the cognitive radio phases. The overhead also reduces the chances of finding a schedule that meets the data exchange requirements. At the same time, the time dedicated to the cognitive radio mechanism prevents the system from entering an idle state with lower energy consumption. The cognitive radio mechanism also requires hardware that can sense different channels, which increases cost and complexity. Additional complexity also arises from the software required to control the protocol.

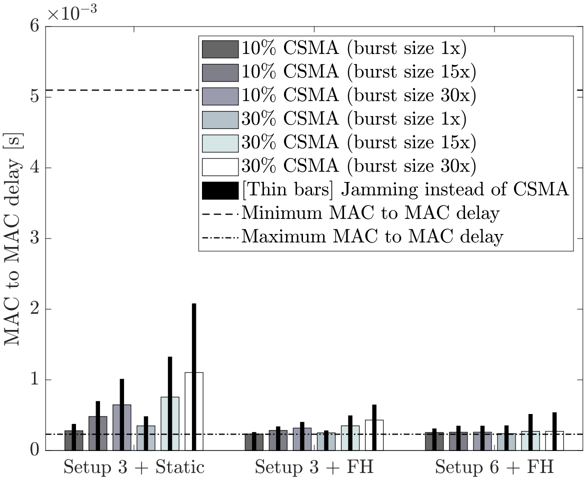


Figure 8.21: Fault prevention using cognitive radio: Average MAC-to-MAC delay for `MAC_CR_None` combined with `MAC_TD_Sprd` (Setup 3) under static and FH interference, and `MAC_CR` combined with `MAC_TD_Sprd` (Setup 6) under FH interference. The wide bars correspond to CSMA interference, whereas the thin black bars represent the same underlying configuration but for jamming interference.

Chapter 9

Contribution area 3: fault-tolerance mechanisms for wireless and wired networks

9.1 Introduction

The fault-prevention mechanisms for the wireless medium proposed in Contribution area 2 (Chapter 8) can effectively curb the occurrence of faulty transmissions. However, even if their number is reduced, faults are still possible. For example, path loss, shadowing, multipath fading and interference still affect wireless transmissions, although hopefully to a lesser extent. Further, in wired environments, a broken cable can lead to a permanent fault if no alternative path is provided for the data. The broken cables could be caused by physical wear, but also by accidents or sabotage. Considering these problems, Contribution area 3 presents six mechanisms that complement the mechanisms from Contribution area 2 to deal with the presence of faults, i.e., fault-tolerance mechanisms, so that they do not lead to failures. The fault-tolerance mechanisms are realized by redundancy in order to achieve higher reliability when individual components can hardly be improved. The proposed mechanisms take two different approaches depending on the transmission medium. The first approach, which is based on time diversity, uses retransmissions to deal with faults that occur in the wireless segments of the network, while the second approach, which is based on space diversity, uses a redundant wireless network to deal with faults that occur in the wired segment.

Being compatible with the mechanisms from Contribution area 2, Contribution area 3 also proposes a hybrid wired and wireless network, where the wired segments rely on the Ethernet-based TTE and TSN technologies and the wireless segments are based on IEEE 802.11. The mechanisms in this contribution area are again compatible with both TTE and TSN, but one of the two should be chosen at the time of deployment.

The physical and data-link layers of the wireless segments are covered by IEEE 802.11 COTS hardware. As stated in Contribution area 2, the technologies for both segments enable high-throughput operation. The applicable topology is multi-hop switched Ethernet, where IEEE 802.11 can be used on the links leading to the end systems.

As in Contribution area 2, the MAC follows a TDMA scheme where each slot has a designated purpose, e.g., for transmissions corresponding to a specific flow or for protocol-related transmissions. The scheduler is responsible for allocating the time slots after considering the data exchange requirements, protocol-related factors, and a multi-hop topology that might require transfers across the wired and wireless segments. Data transfers with different requirements are supported by adopting the TTE traffic classes TT, RC and BE, or the TSN mechanisms ST, CBS and legacy Ethernet traffic. BE traffic can participate in the network, but the fault-tolerance mechanisms are not aimed at handling it.

The proposed MAC mechanisms are summarized in Table 9.1. The faults identified in the problem formulation in Section 7.2.2 which are addressed by the proposed MAC mechanisms are summarized in Table 9.2. Finally, the research questions described in Section 1.3 that are addressed by the proposed MAC mechanisms are included in Table 9.3.

Table 9.1: Summary of the MAC mechanisms proposed in Contribution area 3.

Group	**Label**	**Description**
Fault tolerance using time diversity (described and analyzed in Section 9.3)	`MAC_TD_Cons_1`	Retransmissions placed right after the transmission. No feedback is used, all retransmissions take place.
	`MAC_TD_Cons_2`	Retransmissions placed right after the transmission. Feedback is used, retransmissions only if needed.
	`MAC_TD_Sprd_1`	Retransmissions placed at some distance from the original transmission. No feedback used, all retransmissions take place.
	`MAC_TD_Sprd_2`	Retransmissions placed at some distance from the original transmission. Feedback used, retransmissions only if needed.
Fault tolerance using space diversity (described and analyzed in Section 9.4)	`MAC_SD_Sta`	Static redundancy (hot standby). The wireless backup network is always used for a subset of the data traffic of the wired segment.
	`MAC_SD_Dyn`	Dynamic redundancy (cold standby). The wireless backup network is only used for traffic that was originally scheduled to go through currently broken wired paths.

Table 9.2: Faults (described in Section 7.2.2) addressed by the MAC mechanisms proposed in Contribution area 3. An “x” in the table means that the fault is handled by the MAC mechanism.

MAC mechanism	`MAC_TD_*`	`MAC_SD_*`
Fault 1: delayed transmission due to lack of senders coordination (Table 7.1)	x	
Fault 2: lost transmission due to path loss, shadowing, multipath fading, overlapping transmissions and/or interference (Table 7.2)	x	
Fault 3: lost transmission due to broken wires (Table 7.3)		x

Table 9.3: Research questions (described in Section 1.3) addressed by the MAC mechanisms proposed in Contribution area 3. An “x” in the table means that the research question is addressed by the MAC mechanism.

MAC mechanism	`MAC_TD_*`	`MAC_SD_*`
RQ2.2: What fault-tolerance mechanisms can be used to support reliable and real-time communications for systems with integrated wired and wireless connectivity?	x	x

Note that although the mechanisms proposed in contribution areas 2 and 3 have been evaluated individually in most cases, they can also be combined if and when required by the requirements of the applicable scenarios.

9.2 Related work

9.2.1 Time diversity

Transmitting the same data at different times is a common measure to improve reliability in the literature on dependable and real-time networks. However, these works share a common concern: Additional transmissions are limited by the real-time deadlines of data traffic. Most works propose strategies in which retransmissions are bound to given deadlines. Therefore, the effect of retransmissions under different real-time scheduling policies is also a common problem in the discussed literature.

The work in [127] attempts to answer the question of how many retransmissions are needed in IEEE 802.11 networks that suffer from the typical interference levels of an industrial factory environment. The work concludes that delivery is generally achieved with only one retransmission and that small frames require fewer retransmissions.

The authors in [128] perform an analysis of real-time schedulability using EDF to guarantee that retransmissions are made without deadline misses in scenarios with error rates typically found in wireless communications.

The work in [129] addresses the problem of increasing reliability through retransmissions by comparing several scheduling options based on heuristics in a TDMA protocol that includes relaying nodes that convey information to a central node. Thus, the data exchange interactions in this work are limited to sending to a single node.

The authors in [130] solve the scheduling problem by providing a probabilistic provisioning of retransmissions, where each node computes the number of retransmissions needed for the given channel error model and sends the requirements in a message to the scheduler. Retransmissions can either be immediate or deferred. The mechanism relies on HCCA and a priority-based scheduler that uses a traffic identifier as priority.

Similarly, [131] presents a MAC protocol for WSN where the number of retransmissions is calculated based on the probability of interference and the desired reliability level.

The authors in [132] target reliability in WSN with typical error rates of industrial channels by proposing a TDMA protocol where multiple sensor samples are sent in a message at once and, if the transmission fails, the message is queued to be sent later. However, since the message can be queued indefinitely, the mechanism does not provide real-time guarantees. Therefore, the authors propose an alternative based on forcing all retransmissions to happen inside a longer time slot, an approach that ignores the fact that interference often occurs in bursts.

The work in [80] uses an EDF scheduling policy where retransmissions occur observing deadlines. At runtime, non-requested retransmission opportunities are given to failed transmissions that have not yet missed their deadlines.

In [82], retransmission time slots are pre-allocated in structures cyclically repeated, but the allocation is fixed and inflexible. The time slots can be claimed by other traffic types if the retransmissions are not needed.

The authors in [133] propose two retransmission schemes, one in which the retransmissions occur immediately after the original transmission, and another in which other transmissions can occur between the original transmission and the retransmission, with priority given to the one closest to the deadline. Additional techniques for allocating retransmissions are proposed by the same authors in [134].

The work in [135] proposes retransmissions right after the original transmission in WSANs.

Unfortunately, the aforementioned scheduling solutions including retransmissions are quite limited regarding the scenarios they can handle and cannot be applied in hybrid wired/wireless multi-hop networks.

9.2.2 Space diversity with distinct physical layers

Only a few works were found that combine diverse physical layers to achieve redundancy. The fault-tolerant solution in [136] is conceived for low-energy consumption and low throughput, and relies on the wired and wireless CAN and IEEE 802.15.4 standards, respectively. The nodes can only be arranged in clusters and data from the cluster is collected regularly. The data is then sent to other clusters until the final destination is reached.

The work in [137] proposes a strategy where the data can be transmitted via wired or wireless links depending on the logic of the routing algorithm.

In the approach followed by [138], the end systems establish direct wireless connections after the original wired connections break. The end systems are provided with wireless capabilities based on traffic demand. The work mainly focuses on the routing protocol that is in charge of selecting the path for the transmissions.

9.3 Fault tolerance using time diversity (MAC_TD_Cons_1, MAC_TD_Cons_2, MAC_TD_Sprd_1 and MAC_TD_Sprd_2)

9.3.1 System model

Topology

The mechanisms proposed for handling faults in the wireless segment use exactly the same topology model as the one presented in Contribution area 2. The details about the topology model are provided in Section 8.3.1.

Traffic

The characteristics of the data transmissions in the network where the fault-tolerance mechanisms for wireless networks are applied are the same as the ones presented in Contribution area 2. Section 8.3.2 contains a detailed description of the supported traffic. However, even though three traffic classes are considered in this description, the fault-tolerance mechanisms only target TT/ST and RC/CBS traffic, and leave out BE traffic.

9.3.2 Mechanisms description

The mechanisms proposed in this contribution to cope with faults in the wireless segment are based on time diversity and use retransmissions [4]. The main goal is to increase the reliability of critical data exchanges (as initially indicated in Table 7.8 and Table 7.9 in the fault-handling proposals).

The retransmissions are carried out according to different approaches, resulting in four different options, which are presented below. The generated schedules, which take into account the handling of all traffic classes and retransmissions, are distributed to the participant nodes before the network is put into operation. A detailed description of the scheduler for handling retransmissions of critical traffic is given later in Section 9.3.3.

Retransmission schemes

An essential factor to be considered in the design of retransmission schemes is that interference can occur in bursts characterized by a duration of T_{burst}. In view of this, two retransmission schemes are proposed: One that schedules retransmissions immediately after the corresponding transmission slot, i.e., consecutive retransmissions, labeled as MAC_TD_Cons, and another that sets a gap between the transmission and the retransmissions, i.e., spread retransmissions, labeled as MAC_TD_Sprd. The former is expected to have lower delivery jitter. In contrast, the latter is expected to be better protected against interference bursts. In both cases, a time slot allocated to the critical traffic either for the original transmission or a retransmission can be reused by the BE traffic. Such reuse occurs if no critical traffic is transmitted in the slot and then after the BE message sender has contended to access the channel. An example of the proposed mechanisms is shown in Figure 9.1.

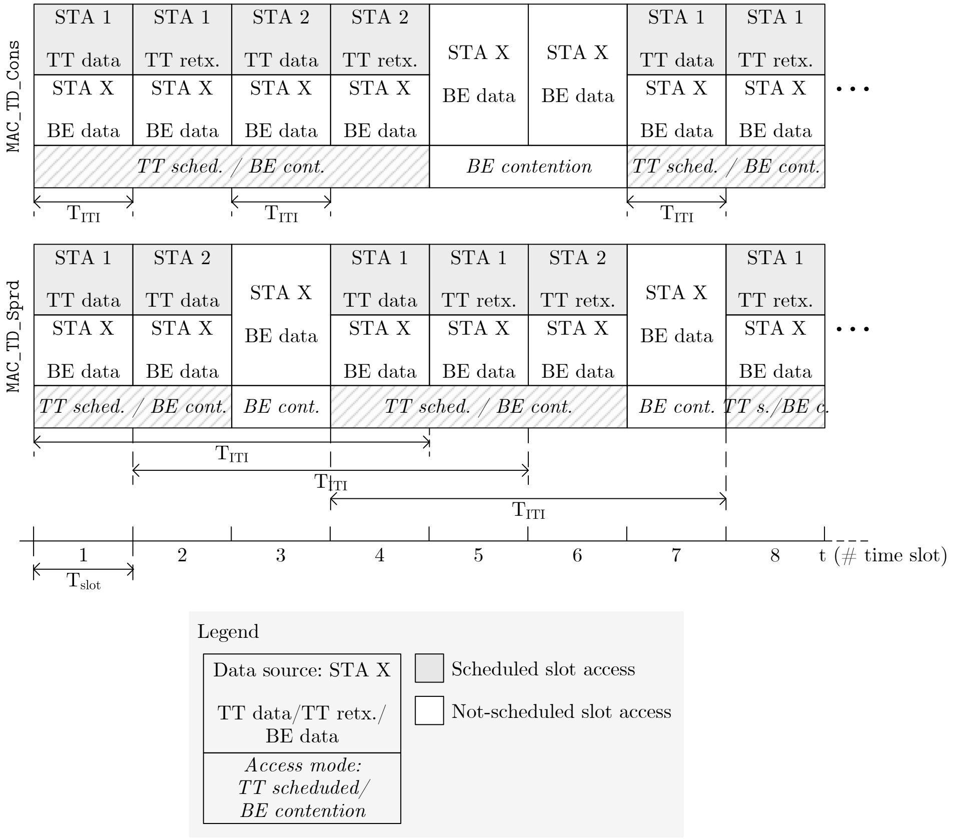


Figure 9.1: Fault tolerance using time diversity: example of time allocation

Two parameters characterize the time diversity mechanisms for a critical message: the number of times $m_i.n$ that a message m_i is transmitted, and the time difference between two consecutive transmissions of the same message instance $m_{i,j}$, named inter transmission interval T_{ITI}, which is the same for all messages $m \in M$. Since the reliability can vary significantly between links, $m_i.n$ is redefined to be expressed per link, so that it refers to the number of times m_i is transmitted on a link $[v_j, v_k]$, yielding $m_i^{[v_j,v_k]}.n$. Such a redefinition makes it possible, for example, to increase the number of transmissions of a message on the less reliable links. Both $m_i^{[v_j,v_k]}.n$ and T_{ITI} can be adjusted to combat interference. In particular, adjusting T_{ITI} to the size of the interference burst T_{burst} so that $T_{ITI} = T_{burst}$, and $m_i^{[v_j,v_k]}.n$ to the level of interference is expected to have a positive effect on reliability. The selection of different values for T_{ITI} and $m_i^{[v_j,v_k]}.n$ has an impact on the delivery latency. The minimum value for the delivery latency happens if and when the first transmission is successful. The maximum delivery latency occurs when the last allowed retransmission is successful. If the transmission is not successful, the delivery latency is considered infinite.

Consecutive retransmissions (MAC_TD_Cons). A straightforward way to include retransmissions is to place them right after the transmission. The mechanism works as shown in Figure 9.1, where the transmission and retransmission are placed in separate time slots. The scheduler first places the transmission and then the retransmissions, with a shift between them of the time slot duration: $T_{ITI} = T_{slot}$.

Spread retransmissions (MAC_TD_Sprd). In this scheme, which is shown in Figure 9.1, the retransmissions are distributed in such a way that the time between the transmission and the retransmission is larger than in the MAC_TD_Cons scheme. The specific value of T_{ITI} must be a compromise between two factors. On the one hand, it is generally easier to obtain a schedule if the value is as low as possible, since the message cannot meet its deadline if the value is too large. A low value also reduces the transmission jitter. On the other hand, a sufficiently large value is expected to make more likely to avoid bursts of interference: $T_{ITI} > T_{burst}$. In this regard, it is beneficial to characterize the interference in each scenario so that parameters such as the maximum interference burst size can be estimated. It should also be noted that while low jitter is very important for TT/ST, this is not necessarily the case for RC/CBS traffic, suggesting that different strategies could be used for TT/ST and RC/CBS traffic.

Slot size and use of feedback mechanism

The benefit of retransmissions is only given if the following two conditions are met: Previous transmission attempts have failed and the data is still useful when retransmitted. Bandwidth is therefore wasted if something is retransmitted after it has been successfully received or if the retransmitted data is no longer valid. For this reason, the sender can be informed about the success of the transmission. The sender can then decide whether to retransmit or not. These mechanisms based on retransmissions after receiving feedback are commonly referred to as ARQ. The feedback is generally conceived as an ACK message, which is sent by the destination when a message arrives without error. The information contained in the ACK message could be used to cancel retransmissions and use this time instead for, e.g., BE traffic or retransmissions from other flows. However, retransmissions always occur in systems without feedback, e.g., in simplex communication systems. In this context, two variants of the MAC mechanisms are introduced: with and without feedback.

Always retransmit (MAC_TD_Cons_1 and MAC_TD_Sprd_1). In the first variant of the time diversity mechanisms, it is proposed not to use feedback, so that the time slots are allocated in advance and retransmissions always take place, as is the case with the scheduled transmissions in WirelessHART. This MAC variant is labeled as

`MAC_TD_Cons_1` or `MAC_TD_Sprd_1`, depending on whether it is applied to consecutive or spread retransmissions, respectively.

The handling of retransmissions without feedback is also described by the pseudocode in Algorithm 9.1. The procedures from the pseudocode are clarified in Appendix A. The pseudocode shows how the time slots are allocated in advance and retransmissions always take place as long as there is data to retransmit (lines 13-16).

Algorithm 9.1 Fault tolerance using time diversity: Pseudocode describing the MAC mechanisms `MAC_TD_Cons_1` and `MAC_TD_Sprd_1`.

```
 1: procedure MAC_TD_1
 2:     $aifs\_prio \leftarrow AIFS\_AC\_VO$
 3:     $aifs\_be \leftarrow AIFS\_AC\_BE$
 4:     while true do
 5:         wait_for_assigned_slot()
 6:         if is_current_slot_critical() then
 7:             if not is_critical_queue_empty() then
 8:                 $m$ ←dequeue_critical()
 9:                 send_csma_ca($m$, $aifs\_prio$, 0)
10:                 enqueue_critical_retx($m$)
11:             else if not is_be_queue_empty() then
12:                 send_be($aifs\_be$, 0)
13:         else if is_current_slot_critical_retx() then
14:             if not is_critical_retx_queue_empty() then
15:                 $m$ ←dequeue_critical_retx()
16:                 send_csma_ca($m$, $aifs\_prio$, 0)
17:             else if not is_be_queue_empty() then
18:                 send_be($aifs\_be$, 0)
19:         else if is_current_slot_be() then
20:             if not is_be_queue_empty() then
21:                 send_be($aifs\_be$, 0)
```

The slot size T_{slot} required to implement this option only needs to accommodate the data transmission time T_{data} of a message and the slot time T_{ST}, where T_{ST} is selected as a guard interval before each transmission to avoid collisions due to propagation delays. The specific value for T_{ST} is taken from the IEEE 802.11 physical layer. The slot size is then calculated as $T_{slot} = T_{ST} + T_{data}$. Figure 9.2 provides a visual representation of the slot size calculation for the proposed MAC mechanisms.

Retransmit if needed (`MAC_TD_Cons_2` and `MAC_TD_Sprd_2`). In the second variant of the time diversity mechanisms, feedback is used so that retransmissions do not take place if the previous message transmission or retransmissions were successful. This MAC variant is labeled as `MAC_TD_Cons_2` or `MAC_TD_Sprd_2`, depending on whether it is applied to consecutive or spread retransmissions, respectively. The advantage of this approach is that the use of a retransmission slot can be changed dynamically, from being scheduled for retransmissions of critical data traffic to a contention-access slot that can be used by other traffic such as BE. With this approach, the ACK message is sent in the same slot immediately after the data is received. If

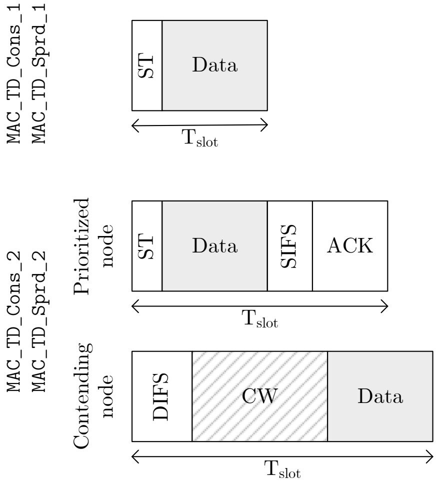


Figure 9.2: Fault tolerance using time diversity: slot size calculations.

the data arrives, it is likely that the channel is still in good condition for the ACK, so the chances of a successful ACK delivery are higher.

The handling of retransmissions with feedback is also described by the pseudocode in Algorithm 9.2. The procedures from the pseudocode are clarified in Appendix A. Some aspects of the pseudocode to emphasize are: The slots intended for the retransmission of critical data traffic (lines 15-17), the contention-access slots (lines 18-19) and the sending of an ACK message immediately after data is received (lines 24-27).

The proposal implies that the size of the slot T_{slot} must have enough space for the two supported uses: Either a prioritized node accesses the slot and uses the feedback mechanism, or the slot is accessed via contention. For the first scenario, the slot must accommodate T_{data}, T_{ST}, the ACK T_{ack} and the short interframe space T_{SIFS}. Both values T_{ST} and T_{SIFS} come from the physical layer of IEEE 802.11. The resulting slot size in the first scenario is $T_{slot} = T_{ST} + T_{data} + T_{SIFS} + T_{ack}$. For the second scenario, both T_{DIFS} and the T_{CW} are required together with T_{data}, resulting in $T_{slot} = T_{DIFS} + T_{CW} + T_{data}$. Overall, T_{slot} is calculated as the maximum between the two options: $T_{slot} = max\{T_{ST} + T_{data} + T_{SIFS} + T_{ack}, T_{DIFS} + T_{CW} + T_{data}\}$. Figure 9.2 provides a visual representation of the slot size calculation for the proposed MAC mechanisms.

The use of DCF mechanisms such as RTS/CTS is not required in the envisaged proposals, but could be introduced if needed by considering slots that can accommodate the additional overhead. Table 9.4 provides a summary of the DCF and ARQ mechanisms that apply to each of the time diversity proposals.

Algorithm 9.2 Fault tolerance using time diversity: Pseudocode describing the MAC mechanisms `MAC_TD_Cons_2` and `MAC_TD_Sprd_2`.

```
procedure MAC_TD_2
    aifs_prio ← AIFS_AC_VO
    aifs_be ← AIFS_AC_BE
    while true do in parallel
        wait_for_assigned_slot()
        if is_current_slot_critical() then
            if not is_critical_queue_empty() then
                m ←dequeue_critical()
                send_csma_ca(m, aifs_prio, 0)
                if not wait_for_ack(m) then
                    enqueue_critical_retx(m)
            else if not is_be_queue_empty() then
                send_be(aifs_be, 0)
        else if is_current_slot_critical_retx() then
            if not is_critical_retx_queue_empty() then
                m ←dequeue_critical_retx()
                send_csma_ca(m, aifs_prio, 0)
            else if not is_be_queue_empty() then
                send_be(aifs_be, 0)
        else if is_current_slot_be() then
            if not is_be_queue_empty() then
                send_be(aifs_be, 0)
    while true do in parallel
        wait_for_msg()
        m ←recv_from_link()
        if is_critical(m) then
            send_ack(m)
```

9.3.3 Scheduling

Scheduling goal

The described fault-tolerance mechanisms for the wireless medium, which handle different types of traffic, require scheduling the utilization of the limited communication resources. The scheduling solution for the fault-prevention in the critical-traffic handling mechanism was introduced in Section 8.4.2. This solution is extended to handle the resources shared by multiple nodes sending and receiving data such that:

- Faults are tolerated using time diversity, enabling the mechanisms `MAC_TD_Cons_1`, `MAC_TD_Cons_2`, `MAC_TD_Sprd_1` and `MAC_TD_Sprd_2`. TT/ST and RC/CBS traffic is scheduled in the wireless segment taking into account the data size, occurrence pattern and delivery deadline requirements. Retransmissions for TT/ST and RC/CBS traffic are also scheduled, leaving a configurable interval between transmissions and retransmissions.

Table 9.4: Fault tolerance using time diversity: summary of DCF and ARQ mechanisms applicable to the proposals. In the table, "CT" stands for critical traffic.

MAC mechanism `MAC_TD_`	`Cons_1`		`Sprd_1`		`Cons_2`		`Sprd_2`	
Traffic type	**CT**	**BE**	**CT**	**BE**	**CT**	**BE**	**CT**	**BE**
$\boldsymbol{T_{DIFS}/T_{AIFS}}$ **sensing required?**	No				No	Yes	No	Yes
Minimum required $\boldsymbol{T_{slot}}$	$T_{ST} + T_{data}$				$max\{T_{ST} + T_{data} + T_{SIFS} + T_{ack}, T_{DIFS} + T_{CW} + T_{data}\}$			
CW required?	No				No	Yes	No	Yes
Backoff required?	No				No	Yes	No	Yes
NAV required?	No							
RTS/CTS required?	No							
ACK required?	No				Yes	No	Yes	No
ARQ required?	No				Yes	No	Yes	No

Integration into the TTE and TSN scheduling solution to support fault tolerance using time diversity

Section 8.4.2 previously described the scheduling constraints for the fault-prevention in the critical-traffic handling mechanism. In order to schedule retransmissions in addition to the original transmissions of critical traffic, the contention-free constraints for the wireless medium are redefined in this contribution as follows:

- **Contention-free constraints for the wireless medium including retransmissions**. These constraints force allocating one or more retransmissions with a customized interval separating them. To define these constraints the network topology and the specification of the data exchanges, including the characteristics of the desired retransmissions, are required.

The previously introduced parameters for tuning the time diversity mechanisms state that $m_i^{[v_k,v_l]}.n$ expresses the number n of transmissions of a message m_i on a link $[v_k, v_l]$, while T_{ITI} models the time between transmissions and retransmissions. Once these have been taken into account, the following equation adds retransmissions to the previously formulated Equation 8.3:

$$
\begin{aligned}
&\forall[v_k, v_l], [v_q, v_r] \in L, \forall m_i, m_j \in M, \\
&\forall a \in \left\{0, 1, ..., \left(\frac{LCM(T)}{T_i} - 1\right)\right\}, \\
&\forall b \in \left\{0, 1, ..., \left(\frac{LCM(T)}{T_j} - 1\right)\right\}, \\
&\forall c \in \{0, 1, ..., m_i^{[v_k, v_l]}.n - 1\}, \\
&\forall d \in \{0, 1, ..., m_j^{[v_q, v_r]}.n - 1\} : \\
&((m_i \neq m_j) \wedge \exists m_i^{[v_k, v_l]} \wedge \exists m_j^{[v_q, v_r]} \\
&\ \wedge (([v_k, v_l], [v_q, v_r]) \in L_{wl}) \Rightarrow \\
&((a \times T_i) + O_i^{[v_k, v_l]} \geq \\
&(b \times T_j) + O_j^{[v_q, v_r]} + T_{slot} + d \times T_{ITI}) \\
&\vee \\
&((b \times T_j) + O_j^{[v_q, v_r]} \geq \\
&(a \times T_i) + O_i^{[v_k, v_l]} + T_{slot} + c \times T_{ITI}).
\end{aligned}
\tag{9.1}
$$

As previously stated for Equation 8.2, it is expected that the message instances $m_{i,x}$ have an offset of one period T_i between them. The same principle applies to $m_{j,x}$ and T_j.

9.3.4 Simulator

Simulator tool selection and details

Computer simulations based on OMNeT++ are used to evaluate MAC_TD_Cons_1, MAC_TD_Cons_2, MAC_TD_Sprd_1 and MAC_TD_Sprd_2. The simulator first introduced for the evaluation of the best-effort handling mechanisms is used and extended in this contribution. A description of why OMNeT++ was chosen as the simulation tool and details on the simulation development can be found in Section 8.5.3.

Simulation goals and features

The aim of the developed simulator is to evaluate the performance of the proposed mechanisms based on selected performance metrics:

- **Fault tolerance using time diversity.** Performance evaluation of critical data transmissions according to the MAC mechanisms MAC_TD_Cons_1, MAC_TD_Cons_2, MAC_TD_Sprd_1 and MAC_TD_Sprd_2, compared to the case where none of these mechanisms is applied (MAC_TD_None[1]). The evaluation is performed in terms of delays, reliability and overhead under different interference patterns using the following performance metrics:

[1] In the performance evaluation, the label MAC_TD_None is used to refer to the scenarios where MAC_TD_Cons_1, MAC_TD_Cons_2, MAC_TD_Sprd_1 and MAC_TD_Sprd_2 are not used.

- **Average delay from MAC layer to MAC layer**. The delay from MAC layer to MAC layer is the time a message takes from entering the MAC layer on the sender side to leaving the MAC layer on the receiver side. This takes into account the possible retransmissions that the message can go through until it is finally received. This performance metric compares the average delay caused by the retransmission schemes, which increases with each failed transmission attempt.
- **Percentage of failed messages**. It is inversely proportional to the service being fulfilled. Since the scheduler guarantees that no retransmissions take place after the message deadline has expired, a deadline miss is accounted for every time a message is not successfully delivered after the last scheduled retransmission. By considering the deadline miss, the behavior of real-time traffic is evaluated.
- **Percentage of time left for BE traffic**. It evaluates the differences between the MAC mechanisms with regard to the time available for BE traffic. The more time available for BE traffic, the less overhead is caused by the mechanism. The differences between the mechanisms result from the different slot sizes. The performance metric also helps to evaluate whether the option in `MAC_TD_Cons_2` and `MAC_TD_Sprd_2` of reusing retransmission slots when they are not needed is beneficial.

The simulator first introduced for the best-effort handling mechanisms (Section 8.5.3) is extended by the following features:

- Modified IEEE 802.11 data-link layer, in particular the MAC layer, to include the following aspects:
  - Dispatching critical data following the fault tolerance using time diversity mechanisms `MAC_TD_Cons_1`, `MAC_TD_Cons_2`, `MAC_TD_Sprd_1` and `MAC_TD_Sprd_2`.

Simulation toolchain

The simulation toolchain HeNRES, first introduced for the best-effort handling mechanisms (Section 8.5.3), is extended to include the simulation of `MAC_TD_Cons_1`, `MAC_TD_Cons_2`, `MAC_TD_Sprd_1` and `MAC_TD_Sprd_2`. The changes for each of the steps in the toolchain are explained below:

Step 1. Network and traffic generator. No modifications in this contribution.

Step 2. Traffic scheduler. The following features are added in this contribution:

- Models the scheduling problem for periodic critical traffic using FOL constraints as previously described in the scheduling section.

Step 3. Network simulator configuration generator. No modifications in this contribution.

Step 4. Network simulator. The network simulator is extended to include the aspects mentioned in the simulation goals and features section.

Step 5. Results processing. No modifications in this contribution. Further, the log checker tool is not extended and does not cover `MAC_TD_Cons_1`, `MAC_TD_Cons_2`, `MAC_TD_Sprd_1` and `MAC_TD_Sprd_2`.

9.3.5 Performance evaluation

The aim of the following experiments campaign is the performance evaluation through simulations of critical data transmissions according to the MAC mechanisms `MAC_TD-_Cons_1`, `MAC_TD_Cons_2`, `MAC_TD_Sprd_1` and `MAC_TD_Sprd_2`, compared to the case where none of these mechanisms is applied, `MAC_TD_None`. The evaluation is performed in terms of delays, reliability and overhead for different interference patterns.

Scenario description

The setup used to compare different configurations to evaluate their performance is a small wireless network with five end nodes and one AP connected according to the topology model (Section 9.3.1). The setup is considered large enough to obtain results that make it possible to identify the differences between the configurations.

As the aim is to support the highest possible transmission speeds, the highest bitrate R offered by IEEE 802.11b is selected, i.e., 11 Mbps. The maximum number of transmission copies allowed for any message m_i on any dataflow link $[v_j, v_k]$, $m_i^{[v_j,v_k]}.n$ is set to 2 (1 transmission and 1 retransmission), as this shortens the simulation time while still showing the necessary performance differences. In addition, the results can easily be extended to a larger number of retransmissions. Table 9.5 provides a summary of the protocol-related parameters.

Traffic travels from the end nodes to the AP and is evenly distributed among the nodes. The size of the exchanged messages $m_i.size$ is 62 bytes, of which 4 bytes are payload, which is relatively small but representative of the messages typically exchanged in industrial networks. To make the results comparable, all scenarios are tested under the same traffic conditions and adjusted to the protocol that can handle the least amount of traffic to avoid queue overflows. Only TT/ST is generated, and the generation is randomized with a sufficient number of messages to fill the scheduled time slots of the schedule that can handle the least amount of traffic. In addition, the generation is evenly distributed among the nodes. The simulations are run to obtain at least 100 message losses per configuration. The traffic-related simulation parameters are summarized in Table 9.6.

To evaluate the proposed MAC mechanisms, an AWGN channel is considered. The free-space model is selected to simulate the path loss. Different channel conditions are also considered in the evaluation, including different types of interference from CSMA devices and jamming interference, the latter being used to model any other source of unintentional interference. The interference occurs at different intensities and is present 10% and 30% of the total time. The interference is randomly generated but can occur in bursts of different sizes. The size is a multiple of the time required to send a message of size $m_i.size$. The worst case interference scenario is represented by

Table 9.5: Fault tolerance using time diversity: protocol-related simulation parameters.

Parameter	**Value**
MAC mechanisms	MAC_TD_None
	MAC_TD_Cons_1
	MAC_TD_Cons_2
	MAC_TD_Sprd_1
	MAC_TD_Sprd_2
DCF and ARQ characteristics	• $T_{ST} = 20\,\mu$s (IEEE 802.11b) • $T_{SIFS} = 10\,\mu$s (IEEE 802.11b) • $T_{DIFS} = T_{SIFS} + 2T_{ST} = 50\,\mu$s (IEEE 802.11b) • CW = [0,31] slots, $T_{CW} = [0,31]T_{ST} = [0,620]\mu s$ (IEEE 802.11b) • $T_{ack} = 203\,\mu$s • Refer to Table 9.4 for additional details.
Wireless data payload rate R (IEEE 802.11b) [Mbps]	11
Maximum number of transmitted copies $m_i^{[v_j,v_k]}.n$	2

Table 9.6: Fault tolerance using time diversity: traffic-related simulation parameters.

Parameter	**Value**
TT/ST data message size $m_i.size$ [bytes]	62
TT/ST data message generation pattern	At random times and as many messages as the generated schedule coping with the least amount of traffic allows.
Number of transmitted TT/ST data messages $\|M\|$	As many as to have ≥ 100 message losses for each configuration.

jamming interference occupying 30% of the time, a scenario that can be considered extreme, i.e., not many protocols, if any, can cope with this level of interference, but it is chosen to test the limits of the proposed protocols. Table 9.7 provides a summary of the parameters related to fault injection.

The combination of various protocol-related parameters (MAC mechanisms) and fault-injection parameters (type of interference, level of interference and interference burst size) results in 60 unique configurations/scenarios to compare.

Table 9.7: Fault tolerance using time diversity: fault-injection-related simulation parameters.

Parameter	**Value**
Channel model	AWGN Free-space path loss model
Type of interference	CSMA
	Jamming
Level of interference	10%
	30%
Interference burst size (as multiple of $m_i.size$)	1x
	15x
	30x

Results

Figure 9.3 shows the percentage of failed message transmissions for MAC_TD_Cons_1 and MAC_TD_Sprd_1 under different loads and burst sizes of CSMA and jamming interference. Figure 9.4 shows the same type of results, but for MAC_TD_Cons_2 and MAC_TD_Sprd_2. In both figures, the results are compared with the case where no retransmissions occur, MAC_TD_None. It can be seen that the jamming interference type is more harmful than CSMA interference, as CSMA devices back off when they sense the medium as busy. The level of interference causes the expected effect: the more time occupied by the interference, the greater the deadline miss ratio. However, it can be seen that a larger burst size does not always lead to worse results. The reason for this is that, for the same level of interference, a larger burst size limits the interference to larger but less frequent interference intervals. It can be deduced from the figures that MAC_TD_Cons_1 and MAC_TD_Cons_2 behave better with the CSMA type of interference. However, MAC_TD_Sprd_1 and MAC_TD_Sprd_2 are much more advantageous for the jamming case. Unfortunately, the jamming interference pattern is still very harmful even with MAC_TD_Sprd_1 and MAC_TD_Sprd_2, but it can be improved by increasing the number of retransmissions.

Figure 9.5 shows the percentage of the total time that BE transmissions can take when using MAC_TD_Cons_1 and MAC_TD_Sprd_1 under different loads and burst sizes of CSMA interference and jamming interference. Figure 9.6 shows the same type of results, but for MAC_TD_Cons_2 and MAC_TD_Sprd_2. In both figures, the results are compared to the case where no retransmissions take place, MAC_TD_None. The BE slots are allocated when MAC_TD_None applies and retransmission slots are left free if the transmission was successful in a previous attempt. The latter is only the case for MAC_TD_Cons_2 and MAC_TD_Sprd_2. In the case of MAC_TD_None, there is a significant difference between MAC_TD_Cons_1 and MAC_TD_Sprd_1 on the one hand and MAC_TD_Cons_2 and MAC_TD_Sprd_2 on the other, as MAC_TD_Cons_1 and MAC_TD_Sprd_1 consider shorter slots and can allocate a larger number of them. In this particular example, where small messages are transmitted, the overhead caused by

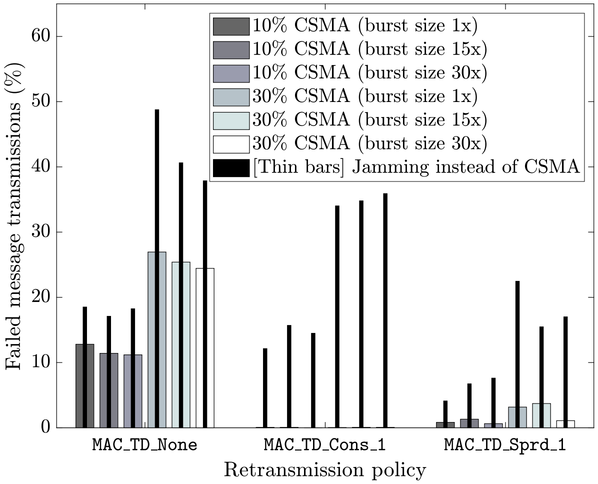


Figure 9.3: Fault tolerance using time diversity: percentage of failed message transmissions for MAC_TD_None, MAC_TD_Cons_1 and MAC_TD_Sprd_1 with different interference levels and interference burst sizes. The wide bars correspond to CSMA interference, while the thin black bars represent the same underlying configuration but with jamming interference.

MAC_TD_Cons_2 and MAC_TD_Sprd_2 is very noticeable and results in MAC_TD_Cons_1 and MAC_TD_Sprd_1 having to allocate almost twice the number of time slots. Better performance for MAC_TD_Cons_2 and MAC_TD_Sprd_2 is expected with an increase in the size of transmitted data.

Figure 9.7 shows the average MAC-to-MAC delay for MAC_TD_None, MAC_TD_Cons-_1 and MAC_TD_Sprd_1 with different interference levels and interference burst sizes of CSMA and jamming interference. Figure 9.8 shows the same type of results, but for MAC_TD_Cons_2 and MAC_TD_Sprd_2. In both figures, the results are compared to the case where no retransmissions take place, MAC_TD_None. In both cases, the delay between the cases MAC_TD_None, MAC_TD_Cons_1 and MAC_TD_Cons_2 does not increase significantly. In the case of MAC_TD_Sprd_1 and MAC_TD_Sprd_2, however, the delay is much greater, as the retransmission time slots are spaced further apart and increase in proportion to the burst size. Note the difference between the average delay for MAC_TD_Cons_1 and MAC_TD_Sprd_1 on the one hand, and MAC_TD_Cons_2 and MAC_TD_Sprd_2 on the other, which is slightly more significant for the latter due to the larger slot size required by the protocol overhead.

9.3.6 Conclusions

Even if the fault-prevention mechanisms from Contribution area 2 help to reduce the occurrence of faults in the communication system, they can still occur. Therefore, this contribution presents the mechanisms MAC_TD_Cons and MAC_TD_Sprd that deal with faults, i.e., fault tolerance mechanisms, by applying redundancy in the time

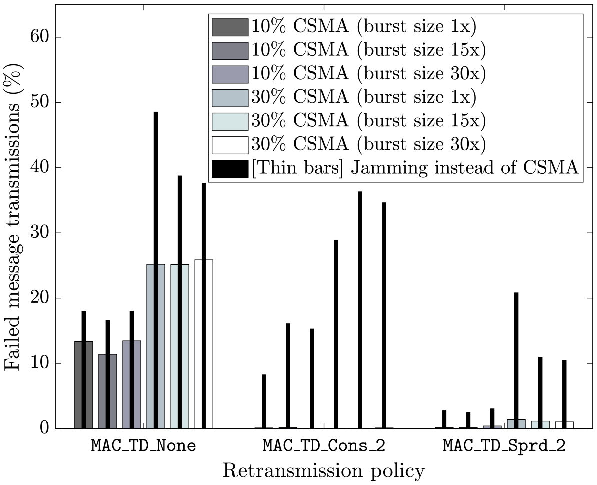


Figure 9.4: Fault tolerance using time diversity: percentage of failed message transmissions for `MAC_TD_None`, `MAC_TD_Cons_2` and `MAC_TD_Sprd_2` with different interference levels and interference burst sizes. The wide bars correspond to CSMA interference, while the thin black bars represent the same underlying configuration but with jamming interference.

dimension. The scheduling solution introduced for `MAC_CH` is extended to include retransmissions. Two variants of MAC mechanisms are investigated, one version that utilizes all scheduled retransmissions and another version that is based on feedback and allows the reuse of the unneeded retransmission slots. Two reliability mechanisms are also applied to these variants, with retransmissions occurring either immediately, `MAC_TD_Cons`, or slightly later, `MAC_TD_Sprd`. The first mechanism minimizes jitter, while the second mechanism copes better with interference bursts. The mechanisms are simulated using the HeNReS simulator in environments exposed to interference of different types, duration and persistence. The results show that retransmissions are an effective way to increase the reliability of the communication system and that the different retransmission schemes can be applied to better cope with interference of different intensity, duration and persistence.

In addition to reliability, availability and safety are also likely to be positively affected by `MAC_TD_Cons` and `MAC_TD_Sprd`. The mechanisms for fault tolerance could make it possible to support highly critical data exchanges. The mechanisms should also help supporting the adaptability to dynamic changes in channel conditions. Determinism depends on the conditions for which it is defined, but should benefit from a higher proportion of successful transmissions. On the negative side, retransmission mechanisms are expected to reduce the available throughput due to the overhead of sending multiple copies of the same data. Such overhead could lead to a preference for shorter and less frequent data exchanges. In addition, greater latency and jitter are expected for the data exchanges. Furthermore, the mechanisms bring more complexity and costs.

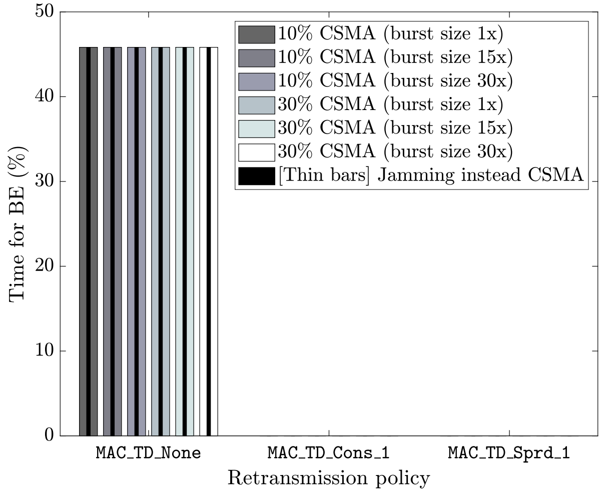


Figure 9.5: Fault tolerance using time diversity: Percentage of time that can be used for BE transmissions when MAC_TD_None, MAC_TD_Cons_1 and MAC_TD_Sprd_1 are applied with different interference levels and interference burst sizes. The wide bars correspond to CSMA interference, while the thin black bars represent the same underlying configuration but with jamming interference.

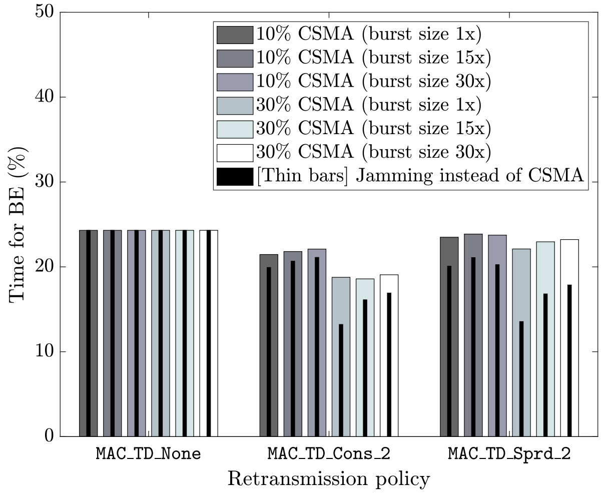


Figure 9.6: Fault tolerance using time diversity: Percentage of time that can be used for BE transmissions when MAC_TD_None, MAC_TD_Cons_2 and MAC_TD_Sprd_2 are applied with different interference levels and interference burst sizes. The wide bars correspond to CSMA interference, while the thin black bars represent the same underlying configuration but with jamming interference.

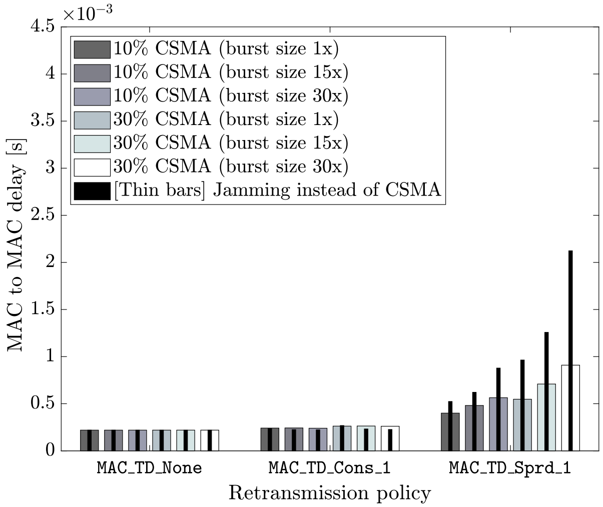


Figure 9.7: Fault tolerance using time diversity: average MAC-to-MAC delay for `MAC_TD_None`, `MAC_TD_Cons_1` and `MAC_TD_Sprd_1` with different interference levels and interference burst sizes. The wide bars correspond to CSMA interference, while the thin black bars represent the same underlying configuration but with jamming interference.

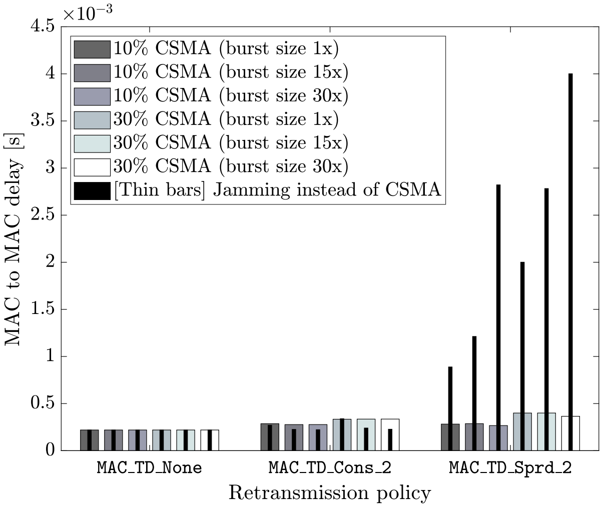


Figure 9.8: Fault tolerance using time diversity: average MAC-to-MAC delay for `MAC_TD_None`, `MAC_TD_Cons_2` and `MAC_TD_Sprd_2` with different interference levels and interference burst sizes. The wide bars correspond to CSMA interference, while the thin black bars represent the same underlying configuration but with jamming interference.

9.4 Fault tolerance using space diversity (MAC_SD_Sta and MAC_SD_Dyn)

9.4.1 System model

Topology

In the realization of the fault tolerance using space diversity mechanisms, the topology also counts with wired and wireless segments, but the use given to them is different than in the topology description for Contribution area 2 (Section 8.3.1). In this case, the focus is on a wired network that connects all nodes. In addition, wireless links are established that do not serve as a primary connection but as a backup. Such a wireless backup attempts to establish a path between two wired network segments if they are disconnected due to faulty wired links.

As in Contribution area 2, the wired network follows the multi-hop switched Ethernet topology found in TTE and TSN. End systems, which act as the source and destination of data, are connected via switches, which act as intermediate nodes and establish a start topology. The switches can also be interconnected, transforming the topology into a star, where each of the leaf nodes can be the central point of another star topology. The links between the devices are full duplex. Although the physical topology can have cycles, the protocols that support Ethernet, e.g., for routing, restrict the logical topologies to avoid cycles that could cause problems such as a broadcast storm.

To enable the use of the wireless backup network, the end systems are also equipped with wireless interfaces. The wireless-capable end systems within range of each other operate in a similar way to a wired bus, i.e., with a single collision domain, as wireless transmissions cannot take place simultaneously on the same frequency. The topology chosen for the wireless network is an IEEE 802.11 ad-hoc topology where wireless end systems can communicate directly with each other. In contrast, the IEEE 802.11 wireless infrastructure topology requires an intermediate node between each pair of end systems. Switches could also equip wireless interfaces, but direct transmission between wireless end systems within range seems a more reasonable option. Using the wireless end systems directly avoids the unnecessary overhead of adding more hops over switches for a message that is going to be transmitted over the wireless medium anyway. In cases where the range of an IEEE 802.11 connection is not sufficient, a routing strategy via wireless switches could be used, but these details are out of the current scope. Figure 9.9 shows an example of the topology, where wired full-duplex connections define two collision domains each, while all wireless-capable nodes are in range and define a single collision domain.

As in Contribution area 2, the physical topology of the multi-hop network is described by the undirected graph $G(V, E)$, where V represents the network nodes (wired and wireless-capable end systems as well as wired switches) and E represents the physical links between the nodes. In the network under consideration, the physical links are bidirectional and are referred to as dataflow links, and the set of dataflow links is referred to as L. Equation 8.1 from Contribution area 2 provides a definition of physical and dataflow links. Further, to simplify the explanation of the upcoming algorithms and figures, the node $v_o \in V$ is used to refer to the wireless broadcast

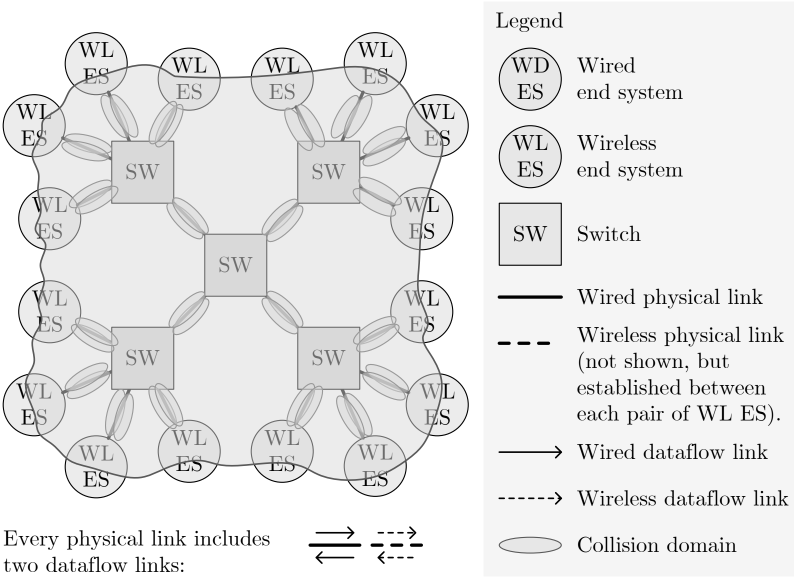


Figure 9.9: Fault tolerance using space diversity: topology model

domain, which means that any link with v_o as destination is able to reach any $v_i \in V$ that has a wireless interface.

Traffic

The characteristics of the data transmissions in the network where the fault-tolerance mechanisms for wired networks are applied are also the same as presented in Contribution area 2. Section 8.3.2 provided a detailed description of the supported traffic. Even though three traffic classes are considered in this description, the fault-tolerance mechanisms only target TT/ST and RC/CBS traffic and leave out BE traffic. In addition to the definitions in Contribution area 2, some further aspects need to be considered.

First, due to the half-duplex connections and limited bandwidth in the wireless segment, it is probably not possible to perform all transmissions happening on the wired segment also on the wireless segment. This means that the static redundancy mechanism introduced later would likely require selecting a subset of all data transmissions that have more demanding requirements on reliability. In TTE, the identification of such a subset could be based on selected VLs, while in TSN it can be based on the stream identification.

In addition to the characterization of traffic presented in Contribution area 2, the fault-tolerance mechanisms for wired networks require a distinction between message types. On the one hand the data messages dm_i and on the other hand the link-state protocol messages lsm, which are used as part of the mechanisms. Data messages dm_i are transmitted periodically over the network, where i is the identifier of the data message and T_i is the period. Each of the periodically generated instances $dm_{i,j}$ is identified by a sequence number j and the pair $\{i, j\}$ is used to detect duplicates. In addition, each dm_i has information about the set of wired paths, $dm_i.paths \in dm.paths$, which is to be transmitted. For the sake of simplicity, it is assumed in these mechanisms that only one wired path exists per dm_i, i.e., $dm_i.path = dm_i.paths$. The actual path is stored in $dm_i.path.path$, which contains the ordered set of dataflow links that it spans in the form $([v_a, v_b], [v_b, v_c], ..., [v_d, v_e])$. Furthermore, the $dm_i.path$ in $dm_i.path.active$ can be in two states indicating whether it is enabled or not.

9.4.2 Mechanisms description

In the wired segments, the problem of broken wires is often avoided by adding additional wired connections, but this comes at a price. Considering this reliability problem, two mechanisms are proposed [6] that deviate from the usual method of adding redundancy in wired networks by replicating the wired network infrastructure. Instead, this contribution explores the space diversity opportunities that arise from using a wireless backup network to restore communications in the event of a broken wire (as initially indicated in Table 7.10 in the fault-handling proposal), assuming that the events that caused the broken wires do not affect the wireless network. The wireless backup is expected to save space, weight and money.

Two mechanisms at the MAC layer are proposed to take advantage of this wireless backup network. The generated schedules considering the handling of all traffic classes are distributed to the participating nodes before the network is put into operation. A description of the scheduling for handling both mechanisms can be found later in Section 9.4.3.

Static redundancy (MAC_SD_Sta)

The static redundancy scheme is based on hot standby, with the scheduled traffic in the wired network also intended for transmission in the wireless network. Apart from some cases where the traffic volume in the wired network is so limited that it can also be accommodated by the wireless network, it is better to assume that not all traffic flows from the wired network can be scheduled in the wireless network, but only a subset of them. As previously mentioned, the different traffic flows can be identified in TTE using the VL encoded in the destination MAC address or the stream identification in TSN.

The static redundancy scheme is described by the pseudocode in Algorithm 9.3. The procedures from the pseudocode are clarified in Appendix A. Further, an example of how it operates is given in Figure 9.10.

The pseudocode also shows the handling of a data message instance $dm_{k,z}$ when working under the static redundancy scheme at node v_i in two cases: as an end system (lines 1-13) or as a switch (lines 14-20). The pseudocode shows how the scheduled traffic in the wired network is also sent in the wireless network (lines 6-9). The pseudocode also includes a mechanism for discarding duplicates in the end systems (lines 11-12). Duplicate detection can be implemented with the sequence number contained in the R-TAG of IEEE 802.1CB in TSN.

Algorithm 9.3 Fault tolerance using space diversity: Pseudocode describing the MAC mechanism `MAC_SD_Sta` from the point of view of end systems (mac_sd_sta_es) and switches (mac_sd_sta_sw).

```
procedure MAC_SD_STA_ES$_i$
    $dm\_recv \leftarrow \{\}$
    while $true$ do
        wait_for_msg()
        if $dm_{k,z}$ ←recv_from_app() then
            for $j \leftarrow 1$ to $|V|$ do
                if $[v_i, v_j] \in dm_k.path.path$ then
                    send_scheduled($dm_{k,z}$, $[v_i, v_j]$)
            send_scheduled($dm_{k,z}$, $[v_i, v_0]$)
        else if $\{dm_{k,z}, [v_x, v_i]\}$ ←recv_from_link() then
            if $dm_{k,z} \ni dm\_recv$ then
                $dm\_recv \leftarrow dm\_recv \cup \{dm_{k,z}\}$
                send_to_app($dm_{k,z}$)

procedure MAC_SD_STA_SW$_i$
    while $true$ do
        wait_for_msg()
        $\{dm_{k,z}, [v_x, v_i]\}$ ←recv_from_link()
        for $j \leftarrow 1$ to $|V|$ do
            if $[v_i, v_j] \in dm_k.path.path$ then
                send_scheduled($dm_{k,z}$, $[v_i, v_j]$)
```

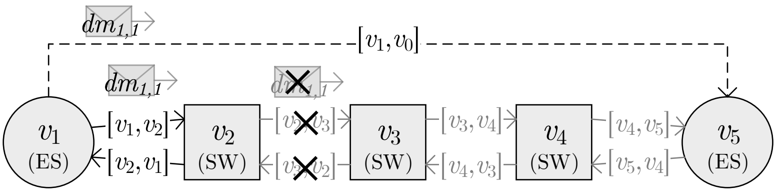


Figure 9.10: Fault tolerance using space diversity: Static redundancy scheme (`MAC_SD-_Sta`). In this example, two paths are used simultaneously to transmit a data message instance$dm_{1,1}$ from v_1 to v_5 via end systems (ESs) and switches (SWs). The wired path (arrows with solid line) breaks, leaving the wireless path (arrow with dashed line) as the only functional.

Dynamic redundancy (`MAC_SD_Dyn`)

The dynamic redundancy scheme is based on the cold standby method and does not pre-allocate the wireless bandwidth. Instead, it reacts to broken paths and establishes an alternative route between the end systems. The dynamic redundancy scheme works in three steps. First, the loss of a wired device or link is detected. A common approach in link-state routing protocols is to periodically send heartbeat messages from each switch to its neighboring switches. After a certain period of time in which no heartbeat has been received, the link can be declared broken. For the sake of simplicity, this contribution assumes that the physical layer of the switches can detect broken links, e.g., by checking the electrical connection.

The dynamic redundancy scheme is described by the pseudocode in Algorithm 9.4. The procedures from the pseudocode are clarified in Appendix A. Further, an example is shown in Figure 9.11.

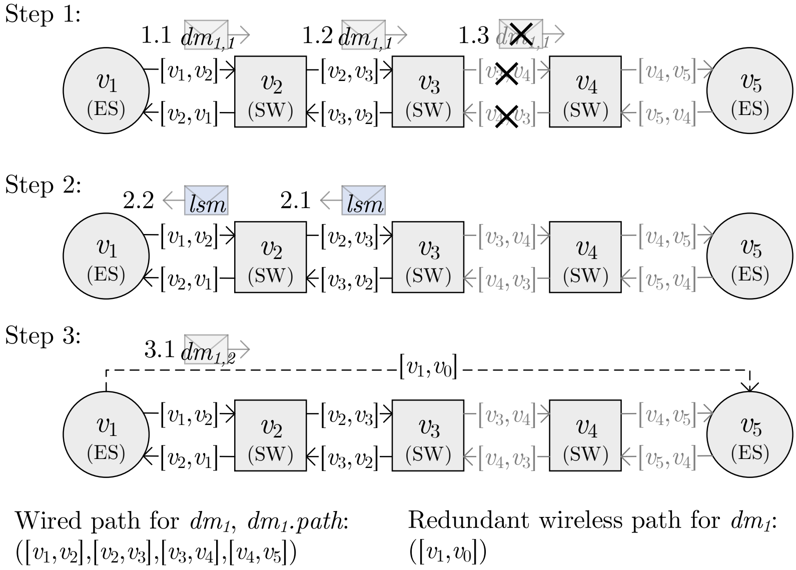


Figure 9.11: Fault tolerance using space diversity: Dynamic redundancy scheme (MAC_SD_Dyn). In this example, the original wired path (solid line arrow) *dm*$_{1}$*.path* for a data message instance $dm_{1,1}$ between v_1 to v_5 via end systems (ESs) and switches (SWs) breaks (step 1). The sender is informed with a link-state message *lsm* (step 2), so that from this point onward it sends the data message instances $dm_{1,x}$ via the wireless interface (dashed line arrow; step 3).

As soon as the sender detects that a link $[v_i, v_j]$ is broken (Algorithm 9.4: lines 10 and 27; Figure 9.11: step 1), the switches having routes traversing the broken link are informed. For this task, scheduled paths could be used that correspond to the reverse of the original paths of the data messages from *dm*$_k$*.paths*. However, for the sake of simplicity and considering that cold standby aims for better bandwidth allocation, any time remaining after the scheduled traffic should be used for these exceptional transmissions. A link-state message *lsm* carrying an identifier for the broken link *lsm.broken_link* ← $[v_i, v_j]$ is sent on all routes that include the broken link $[v_i, v_j] \in$ *dm*$_k$*.path.path* (Algorithm 9.4: lines 28-29; Figure 9.11: step 2). The *lsm* might hop through several switches, with each switch deleting the broken path from its own list of routes and forwarding it via the affected paths (Algorithm 9.4: line 37). An implementation over Ethernet can use the LSP messages from IS-IS to perform the updates in the routing tables. The next time a switch needs to send a data

Algorithm 9.4 Fault tolerance using space diversity: Pseudocode describing the MAC mechanism `MAC_SD_Dyn` from the point of view of end systems (mac_sd_dyn_es) and switches (mac_sd_dyn_sw). The pseudocode also describes the procedure for sending a LSM message (send_lsm).

```
procedure MAC_SD_DYN_ES$_i$
    $dm\_recv \leftarrow \{\}$
    for all $dm_k.path \in dm.paths$ do
        $dm_k.path.active \leftarrow true$
    while $true$ do
        wait_for_msg()
        if $df_{k,z}$ ←recv_from_app() then
            for $j \leftarrow 1$ to $|V|$ do
                if $[v_i, v_j] \in dm_k.path.path$ then
                    if check_phy($[v_i, v_j]$) then
                        $dm_k.path.active \leftarrow false$
                    if not $dm_k.path.active$ then
                        send_asap($dm_{k,z}$, $[v_i, v_0]$)
                    else
                        send_scheduled($dm_{k,z}$, $[v_i, v_j]$)
        else if $\{dm_{k,z}, [v_x, v_i]\}$ ←recv_from_link() then
            if $dm_{k,z} \ni dm\_recv$ then
                $dm\_recv \leftarrow dm\_recv \cup \{dm_{k,z}\}$
                send_to_app($dm_{k,z}$)
        else if $\{lsm, [v_x, v_i]\}$ ←recv_from_link() then
            for $j \leftarrow 1$ to $|dm|$ do
                if $lsm.broken\_link \in dm_j.path.path$ then
                    $dm_j.path.active \leftarrow false$
procedure MAC_SD_DYN_SW$_i$
    while $true$ do in parallel
        for $j \leftarrow 1$ to $|V|$ do
            if check_phy($[v_i, v_j]$) then
                $lsm.broken\_link \leftarrow [v_i, v_j]$
                send_lsm($lsm$)
    while $true$ do in parallel
        wait_for_msg()
        if $\{dm_{k,z}, [v_x, v_i]\}$ ←recv_from_link() then
            for $j \leftarrow 1$ to $|V|$ do
                if $[v_i, v_j] \in dm_k.path.path$ then
                    send_scheduled($dm_{k,z}$, $[v_i, v_j]$)
        else if $\{lsm, [v_x, v_i]\}$ ←recv_from_link() then
            send_lsm($lsm$)
procedure SEND_LSM($lsm$)
    for all $dm_k.path \in dm.paths$ do
        for $j \leftarrow 1$ to $|dm_k.path.path|$ do
            if $lsm.broken\_link = dm_k.path.path_j$ then
                send_asap($lsm$, $dm_k.path.path_{j-1}$)
```

message, it uses the other redundant paths, if available, instead of the path that just broke (Algorithm 9.4: line 13; Figure 9.11: step 3). Since access to the wireless backup network only occurs after random situations such as the one described, the wireless transmission slots are not assigned in advance. Instead, the standard CSMA/CA protocol is used, even if it might cause collisions between transmissions.

The pseudocode shows the handling of a data message instance $dm_{k,z}$ and of link-state messages lsm at node v_i in the case that this is an end system (Algorithm 9.4: lines 1-23) or a switch (Algorithm 9.4: lines 24-37) if they operate according to the dynamic redundancy scheme and including the same mechanism for discarding duplicates at the end systems from the static redundancy case (Algorithm 9.4: lines 17-18).

Both MAC_SD_Sta and MAC_SD_Dyn do not take transient faults into account for the wired links, i.e., once they break, they cannot be recovered.

The use of DCF mechanisms such as RTS/CTS is not required in the envisaged proposals, but could be introduced if necessary by considering slots long enough to accommodate the additional overhead. Sensing the channel and having CW could be helpful to reduce the probability of transmissions starting at the same time. In addition, ARQ is not required as wireless transmissions on the backup channel only have one chance of being transmitted. Table 9.8 summarizes the DCF and ARQ mechanisms that apply to each of the space diversity proposals.

Table 9.8: Fault tolerance using space diversity: summary of DCF and ARQ mechanisms applicable to the proposals.

MAC mechanism	MAC_SD_Sta	MAC_SD_Dyn
T_{DIFS}/T_{AIFS} sensing required?	Yes	
Minimum required T_{slot}	Non-slotted schedule.	
CW required?	Yes	
Backoff required?	No	
NAV required?	No	
RTS/CTS required?	No	
ACK required?	No	
ARQ required?	No	

9.4.3 Scheduling

Scheduling goal

The described fault-tolerance mechanisms for the wired medium, using a wireless backup network and handling different types of traffic, require scheduling the utilization of limited communication resources. The scheduling solution for the fault-prevention in the critical-traffic handling mechanism was introduced in Section 8.4.2. This solution is extended to handle the resources shared by multiple nodes sending and receiving data such that:

- Faults are tolerated using space diversity, enabling the mechanisms MAC_SD_Sta and MAC_SD_Dyn. TT/ST and RC/CBS traffic are scheduled in the wired network considering data size, occurrence pattern and delivery deadline requirements. In the wireless backup network, MAC_SD_Sta mirrors a subset of the transmissions sent by the source nodes in the wired network. Since the transmissions are mirrored, the formulation of the schedule does not need to be changed. In the case of MAC_SD_Dyn, the transmissions are performed with CSMA/CA and do not require scheduling.

Integration into the TTE and TSN scheduling solution to support fault tolerance using space diversity

Section 8.4.2 previously described the scheduling constraints for the fault-prevention in the critical-traffic handling mechanism. The wired network therefore counts with scheduled transmissions for TT/ST and RC/CBS traffic. Since MAC_SD_Sta mirrors a subset of the transmissions from the wired network in the wireless backup network, the scheduler does not need to undergo any modifications. The mirrored transmissions in the wireless network only need to be sent at the same time as the schedule in the wired network sends them from the data source node. As for MAC_SD_Dyn, it does not require scheduling configuration since it is based on CSMA/CA.

9.4.4 Reliability analysis

The analysis of the reliability of the fault-tolerance mechanisms using space diversity is introduced next following one of the techniques previously described in the dependability analysis section (Section 4.2.4). Specifically, the following analysis uses RBD to calculate the reliability of the entire system at the instant t: $R_S(t)$. A data message dm_i has an associated reliability $R_i(t)$, and its influence on $R_S(t)$ is weighted with w_i depending on the amount of data it carries yielding

$$R_S(t) = \sum_{i=1}^{|dm|} w_i R_i(t) \tag{9.2}$$

and the weight w_i being proportional to T_i as in

$$w_i = \frac{T_i}{\sum_{j=1}^{|dm|} T_j}. \tag{9.3}$$

Each dm_i goes over the set of wired dataflow links specified in $dm_i.path.path$, which fail at a constant rate λ_{wd}.

In the basic case without spatial redundancy (MAC_SD_None[2]), the reliability of a data message is calculated as the reliability of a set of blocks, i.e., wired links, connected in series:

[2] In the reliability analysis and the performance evaluation, the label MAC_SD_None is used to refer to the scenarios where MAC_SD_Sta and MAC_SD_Dyn are not used.

$$R^i_{none}(t) = R^i_{wd}(t) = \prod_{k=1}^{|dm_i.path.path|} R_k = \prod_{k=1}^{|dm_i.path.path|} e^{-\lambda_{wd} t}. \tag{9.4}$$

When using static redundancy (MAC_SD_Sta), a selected number of data messages are transmitted via a wireless dataflow link. Note that Equation 9.4 still applies to all other data messages that do not use static redundancy. With static redundancy, dm_i is successfully transmitted if either or both the wired or the wireless paths work. For simplicity, a wireless transmission between two devices in range will fail if it collides with other transmissions. Since the static allocation avoids simultaneous transmissions, this results in a reliability value of 1:

$$R^i_{sta}(t) = 1 - (1 - R^i_{wd}(t))(1 - R^i_{sta\_wl}(t)) = \\ 1 - \left(1 - \prod_{k=1}^{|dm_i.path.path|} e^{-\lambda_{wd} t}\right)(1 - 1) = 1. \tag{9.5}$$

As with dynamic redundancy (MAC_SD_Dyn), dm_i is transmitted via the wireless path if the wired path breaks ($dm_i.path.active = false$):

$$R^i_{dyn}(t) = 1 - (1 - R^i_{wd}(t))(1 - R^i_{dyn\_wl}(t)) = \\ 1 - \left(1 - \prod_{k=1}^{|dm_i.path.path|} e^{-\lambda_{wd} t}\right)(1 - R^i_{dyn\_wl}(t)). \tag{9.6}$$

The reliability of the dm_i transmitted via the wireless path in the case of dynamic redundancy depends on it not colliding with other wireless paths:

$$R^i_{dyn\_wl}(t) = \prod_{j=1, j \neq i}^{|dm_i.path.path|} R^{i,j}_{not\_col}(t). \tag{9.7}$$

The probability that dm_i does not collide with another data message dm_j over the wireless channel depends on the probability that dm_j has a working wired path ($dm_j.path.active = true$) or, if this is not the case, that it does not collide on the wireless channel:

$$R^{i,j}_{not\_col}(t) = R^j_{wd}(t) + (1 - R^j_{wd}(t)) \cdot R_{not\_col\_wl}(t). \tag{9.8}$$

Finally, [139] provides an estimate of the probability that a wireless transmission under DCF will not collide, given the total number of wireless systems that might attempt to transmit, which for simplicity in these mechanisms is assumed to be equal to the number of data messages $|dm|$, i.e., each data message is transmitted by a different wireless system, which adds some pessimism to the analysis, yielding

$$R_{not\_col\_wl}(t) = (1 - \tau)^{(|dm|-1)} \tag{9.9}$$

where τ is the channel access probability of a single wireless system during the slot time T_{ST}, i.e., the time required for a wireless system to detect ongoing transmissions from other wireless systems, a value that depends on the physical layer used and is given by the IEEE 802.11 standard. Based on the simplification that each dm_i is transmitted by a different wireless system, the channel access probability of a wireless system v_i is proportional to its period T_i, so that τ is calculated as the average value for the channel access probability of all wireless systems:

$$\tau = \frac{\sum_{i=1}^{|dm|} \frac{T_{ST}}{T_i}}{|dm|}. \tag{9.10}$$

9.4.5 Simulator

Simulator tool selection and details

Computer simulations based on OMNeT++ are used to evaluate `MAC_SD_Sta` and `MAC_SD_Dyn`. The simulator first introduced for the evaluation of the best-effort handling mechanisms is used and extended in this contribution. A description of why OMNeT++ was chosen as the simulation tool and details on the simulation development can be found in Section 8.5.3.

Simulation goals and features

The aim of the developed simulator is to evaluate the performance of the proposed mechanisms based on selected performance metrics:

- **Fault tolerance using space diversity**. Performance evaluation of critical data transmissions over wired networks with a wireless backup network using the MAC schemes `MAC_SD_Sta` and `MAC_SD_Dyn` compared to the case without spatial redundancy (`MAC_SD_None`). The evaluation is done in terms of protocol overhead, reliability and availability in faulty networks where wired links break using the following performance metrics:
  - **Reliability**. Proportion of data messages received in relation to those sent.
  - **Percentage of duplicated data messages**. Overhead introduced by the `MAC_SD_Sta` in the form of duplicate messages received at the destination.
  - **Percentage of colliding data messages**. Shows the collisions caused by wireless transmissions under `MAC_SD_Dyn`. A high number indicates an overloaded wireless medium, which happens after the wired paths start to break and transmissions use the wireless medium accessed via CSMA/CA.

  - **Responsiveness**. Measures the time between when a wired link breaks and the end systems from which the data originated are notified, so that an alternative path can be activated and a communication path is restored after the fault.

The simulator first introduced for the best-effort handling mechanisms (Section 8.5.3) is extended by the following features:

- Modified Ethernet data-link layer, in particular the MAC layer, to include the following aspects that enable the fault tolerance using space diversity mechanisms MAC_SD_Sta and MAC_SD_Dyn:
  - Dispatching critical data according to a time-triggered schedule.
  - A minimal implementation of the IS-IS link-state protocol.
  - A minimal implementation of stream identification and frame replication and elimination for reliability (IEEE 802.1CB).
  - Implementation for switches and wired end systems.
- Implementation of new hybrid end systems that have both wired and wireless interfaces and, in the case of counting with redundant routes, select the one with the lowest cost to perform the transmission. These hybrid end systems enable the fault tolerance using space diversity mechanisms MAC_SD_Sta and MAC_SD_Dyn.
- Mechanisms for fault injection. Added functionality that enables fault injection in the Ethernet network, causing wired links to break at a given rate.

The simulations for MAC_SD_Sta and MAC_SD_Dyn are as well complemented by the reliability analysis based on RBD. Having both approaches serves to compare the theoretical limits from the reliability analysis with the simulated values.

Simulation toolchain

The simulation toolchain HeNRES, first introduced for the best-effort handling mechanisms (Section 8.5.3), is extended to include the simulation of MAC_SD_Sta and MAC_SD_Dyn. The changes for each of the steps in the toolchain are explained below:

Step 1. Network and traffic generator. No modifications in this contribution.

Step 2. Traffic scheduler. The following features are added in this contribution:

- Models the scheduling problem for periodic critical traffic using FOL constraints as previously described in the scheduling section.

Step 3. Network simulator configuration generator. No modifications in this contribution.

Step 4. Network simulator. The network simulator is extended to include the aspects previously mentioned in the simulation goals and features section.

Step 5. Results processing. No modifications in this contribution. Further, the log checker tool is not extended and does not cover `MAC_SD_Sta` and `MAC_SD_Dyn`.

9.4.6 Performance evaluation

The aim of the following experiments campaign is to evaluate through reliability analysis and simulations the performance of data transmissions over wired networks supported by a wireless backup network with the static and dynamic redundancy schemes `MAC_SD_Sta` and `MAC_SD_Dyn` compared to the case without spatial redundancy, `MAC-_SD_None`. The evaluation is performed in terms of reliability, protocol overhead and responsiveness of the mechanisms in faulty networks where wired links break.

Scenario description

For the reliability analysis and simulation, a total of 12 data messages dm_i are generated based on the parameters from Table 9.9. The messages are transmitted over a mid-sized wired multi-hop star topology network according to the paths shown in Figure 9.12. The transmission times of the data messages are scheduled in such a way that collisions are avoided in the wired network. The end systems are wireless-capable and can perform direct ad-hoc transmissions. It is assumed that all end systems are in range when communicating over the wireless network, so that wireless transmissions collide if they occur simultaneously.

Table 9.9: Fault tolerance using space diversity: traffic-related reliability analysis and simulation parameters.

Parameter	**Value**
Data message size $dm_i.size$ [bytes]	512
Data message R-TAG size [bytes]	6
Link-state message size $lsm.size$ [bytes]	720
Data message period T_i (number of flows sharing the same T_i)	1 ms (6x) 2 ms (2x) 4 ms (4x)
Number of transmitted data messages $\lvert M \rvert$	As many as to observe $R_S(t)$ stabilize.

Table 9.10 summarizes the protocol-related parameters of the reliability analysis and simulation. In the case of `MAC_SD_Sta`, a subset of the flows included in Table 9.9 is selected for transmission via the wireless backup network. The rate R for the wired and wireless network is set to 100 Mbps and 54 Mbps, respectively.

The value of the failure rate for wired links λ_{wd} is set to resemble the behavior of a network where the number of broken wired links ranges from 0 to all broken and is not set to represent a real fault rate of specific components. In the simulation, the two dataflows that are part of the same full-duplex wire break simultaneously, but on average the λ_{wd} is preserved. For simplicity and considering that the main objective of the tests is to analyze the improvements of the reliability schemes, it is assumed that

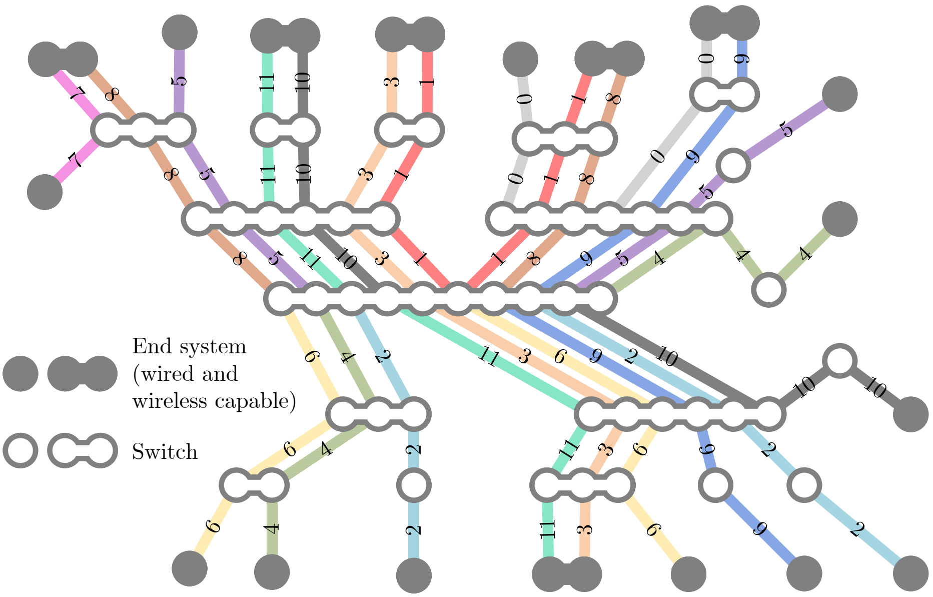


Figure 9.12: Fault tolerance using space diversity: Evaluation network depicting the data message paths. Each of the 12 generated data message paths is identified by a different number.

the nodes do not fail but only the wired links fail. It should be noted that if the end systems fail, they represent a single point of failure in the flows and the reliability of the flows that have them as origin or destination would drop to zero. The wireless links are also assumed to be error-free, with the exception of collisions and the simulated AWGN and free-space path loss model for the channel. Table 9.11 summarizes the reliability analysis and simulation parameters related to fault injection.

The simulation is based on the redundancy schemes model created for OMNeT++. The reliability analysis is carried out with Matlab. The results for the reliability of the overall system $R_S(t)$ are determined for the various scenarios evaluated. Both the simulation and the analysis run until $R_S(t)$ becomes stable, which happens shortly after all wired links break. The choice of λ_{wd} for the wired links directly impacts the time it takes for the simulation to observe trends in the results. In this contribution, the simulation tool enforces the lower limit for λ_{wd}, not in terms of simulation time, but due to the amount of simulation log data, which quickly reaches tens of gigabytes.

The comparison is done for the three different MAC mechanisms. Different values for other parameters are not taken into account, so that a total of three scenarios/configurations are compared.

Results

Figure 9.13 shows the value of $R_S(t)$ for MAC_SD_None compared to when MAC_SD_Sta and MAC_SD_Dyn are applied. The results are given for the simulation and the reliability analysis. The overall pessimistic approximation of the reliability analysis

Table 9.10: Fault tolerance using space diversity: protocol-related reliability analysis and simulation parameters.

Parameter	**Value**
MAC mechanisms	`MAC_SD_None`
	`MAC_SD_Sta`
	`MAC_SD_Dyn`
Data message period T_i of redundant flows in `MAC_SD_Sta` (number of flows sharing the same T_i)	1 ms (1x) 4 ms (1x)
DCF and ARQ characteristics	• $T_{ST} = 9\,\mu$s (IEEE 802.11g) • $T_{SIFS} = 10\,\mu$s (IEEE 802.11g) • $T_{DIFS} = T_{SIFS} + 2T_{ST} = 28\,\mu$s (IEEE 802.11g) • $CW_{min} = 15$ slots, $CW_{max} = 1023$ slots; Minimal $T_{CW} = [0, 15]T_{ST} = [0, 135]\mu s$, Maximal $T_{CW} = [0, 1023]T_{ST} = [0, 9207]\mu s$ (IEEE 802.11g) • Refer to Table 9.8 for additional details.
Wireless data payload rate (IEEE 802.11g) [Mbps]	54
Wired data rate [Mbps]	100

Table 9.11: Fault tolerance using space diversity: fault-injection-related reliability analysis and simulation parameters.

Parameter	**Value**
Channel model	AWGN Free-space path loss model
λ_{wd} [faults/s]	0.00005

becomes clear in comparison to the simulation results. Nevertheless, the lines for reliability analysis and simulation are of the same order of magnitude and follow a similar pattern. If `MAC_SD_None` applies, $R_S(t)$ drops to 0 over time. These results reflect the case where the links in a wired network get broken and there is no other way of transmitting data between the end systems. `MAC_SD_Sta` already achieves a sustained value of $R_S = 0.2$ when all wired links break. The value of $R_S(t)$ depends on the weight w of the data messages that get a chance to transmit over the wireless backup network; in this case they transmit about 20% of the traffic. `MAC_SD_Dyn` keeps a stable value over $R_S = 0.5$, about 30% improvement over `MAC_SD_Sta` in the analyzed case. The limits on the $R_S(t)$ improvement are given by the amount of traffic that the wireless channel can accommodate, so intuitively the $R_S(t)$ of MAC_SD_Sta and MAC_SD_Dyn should be quite similar. However, the collision-free guarantee of MAC_SD_Sta limits the addition of redundant paths, even if they rarely suffer from collisions.

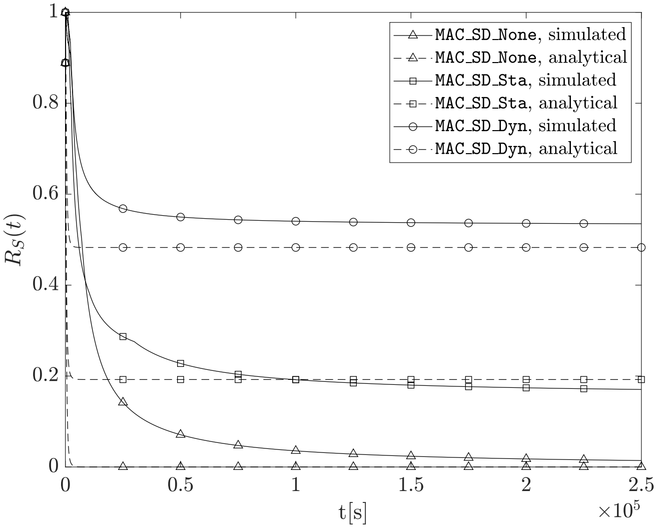


Figure 9.13: Fault tolerance using space diversity: simulated and analytical reliability in the case of MAC_SD_None, MAC_SD_Sta and MAC_SD_Dyn.

Figure 9.14 shows the percentage of duplicated data messages on the receiver end systems, which is determined by simulation. The simulated $R_S(t)$ is added as a reference. The duplicate data messages are the main drawback of MAC_SD_Sta, while there are no duplicates in MAC_SD_None and MAC_SD_Dyn. Duplicates indicate that both wired and wireless transmissions are successful for a data message, although only one of them would have been required. The figure shows that the number of duplicates is the highest at the beginning of the simulation, which corresponds to the proportion of data messages that can be transmitted via both the wired and wireless media. These high numbers quickly decrease when the wired links break, as one broken wired link might affect multiple wired paths. The number of duplicates tends towards 0 as the time approaches when all wired links break. This indicator can be interpreted as the amount of redundancy in the system and how it evolves until it is completely lost, meaning that the system loses its ability to tolerate further faults.

The results from Figure 9.15 show the percentage of colliding data messages when transmitted over the wireless medium. Simulated $R_S(t)$ has been added for reference. This performance measure is interesting for MAC_SD_Dyn, where the wireless backup network is a shared resource for each end system that has detected the wired path as broken. With the traffic volume considered in the simulation, about 40% of the data messages sent over the wireless network collide and do not reach their destination. This number peaks at the beginning of the simulation, as soon as the wired links start

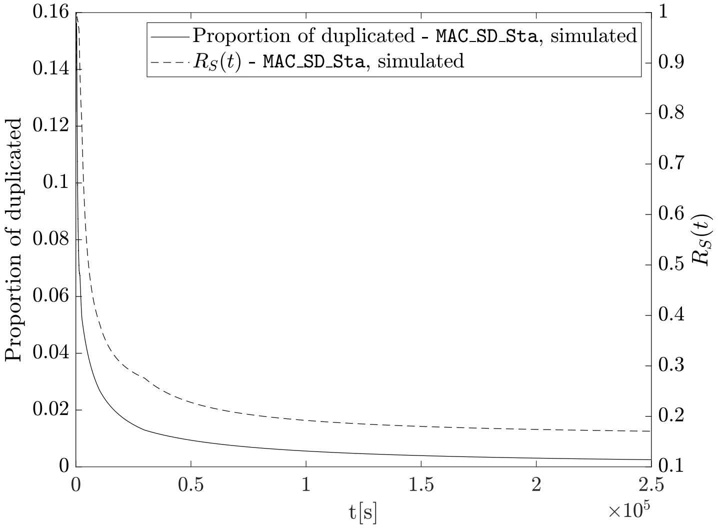


Figure 9.14: Fault tolerance using space diversity: Simulated percentage of duplicate data messages expressed as a decimal compared to simulated reliability. Duplicates only occur when `MAC_SD_Sta` applies.

breaking and stabilizes with the number of wireless transmissions. Despite the high percentage of lost data messages and the resulting impact on $R_S(t)$, `MAC_SD_Dyn` is still better than `MAC_SD_Sta` in the tests performed. However, several techniques can be used to solve the problem of colliding transmissions and improve $R_S(t)$ by better wireless bandwidth allocation than CSMA/CA, as shown by the fault-prevention mechanisms presented in Contribution area 2 (Chapter 8).

One of the advantages of `MAC_SD_Sta` is that recovery in the event of a fault does not take long, as the redundant paths work in parallel. Fortunately, the responsiveness of `MAC_SD_Dyn` is also good. The delay in notifying the sender is in the interval [58.3 μs, 291.4 μs], with an average of $\overline{X}$ =131.2 μs and a standard deviation σ =103.2 μs.

The $R_S(t)$ numbers of `MAC_SD_Sta` are worse in the run experiments than those of `MAC_SD_Dyn`. However, in `MAC_SD_Sta` the data messages are guaranteed to be collision-free. While `MAC_SD_Dyn` could provide a better average $R_S(t)$, there is not a collision-free guarantee. Therefore, the choice of redundancy mechanism depends on what the user prioritizes: a better average performance or a guarantee for the most critical data messages.

9.4.7 Conclusions

In the context of integrated wired and wireless networks, an additional fault is identified, this time involving lost transmissions due to broken wires. In this contribution, two mechanisms for fault tolerance using space diversity `MAC_SD_Sta` and `MAC_SD_Dyn` are proposed by deploying a wireless backup network to increase the reliability of a wired network suffering from broken links. One redundancy mechanism, `MAC-`

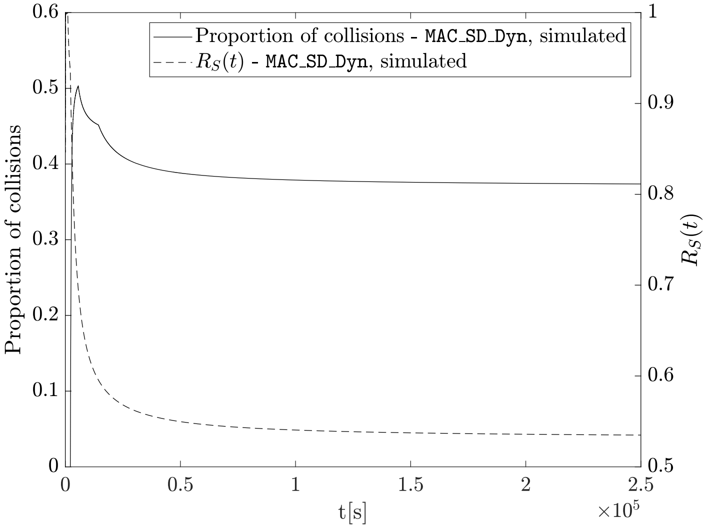


Figure 9.15: Fault tolerance using space diversity: Simulated percentage of wireless collisions expressed as a decimal compared to simulated reliability. Collisions only occur when MAC_SD_Dyn applies.

_SD_Sta, statically allocates communication slots and operates in parallel with the wired network. The mechanism relies the scheduling solution introduced for MAC_CH. The other redundancy mechanism, MAC_SD_Dyn, reacts dynamically to broken wired links and establishes an alternative wireless connection. Reliability is evaluated by reliability analysis and through a simulation using the HeNReS simulator, with each of the methods showing similar behavior in both methods. A constant fault rate in the wired links leads to a gradual degradation in the performance of the wired network. When all wired links break, the MAC_SD_Sta mechanism is able to provide a reliability of 20%, reaching 100% for selected scheduled data messages, while the overhead, expressed as the percentage of duplicated data messages, tends to 0 as a function of time. The MAC_SD_Dyn mechanism is able to deliver more than 50% of the data messages and recover the path almost instantly, although the dynamic allocation of the wireless medium causes a significant number of collisions. The results show that using a wireless backup network as a fault-tolerance mechanism can effectively increase the reliability of a wired network by avoiding the problem of permanent faults. The choice of one of the proposed redundancy mechanisms is subject to a trade-off between ensuring the most critical data messages in the case of MAC_SD_Sta or better average performance in the case of MAC_SD_Dyn.

In addition to reliability, availability and safety can also be positively influenced by MAC_SD_Sta and MAC_SD_Dyn. These mechanisms for fault tolerance could also enable the support for highly critical data exchanges. In addition, the mechanisms are expected to support adaptability to dynamic changes caused by wire breaks, especially in the case of MAC_SD_Dyn. Determinism will depend on the conditions for which it is defined, but should benefit from a higher proportion of successful transmissions. The wireless backup network could also be used to support the synchronization protocol after the wired paths originally used for it break. The available throughput of the wireless backup network is expected to be lower than that of the wired network, which also has a negative impact on large data sizes, frequent occurrence patterns and scalability, but is still better than a broken wired network with the inability to communicate. The wireless backup network could allow the paths between senders and receivers to be shortened compared to the wired network, reducing some of the overhead of performing multi-hop transmissions and lowering delivery latencies. On the downside, this solution requires the additional hardware support of a wireless backup network, which, together with the implementation of the actual mechanisms, makes it more complex and costly.

Chapter 10

Conclusions and future work

In addition to the basic function of exchanging data between different computer systems, data communication networks often have to fulfill a number of other requirements. In the OT field, these often include dependability and real-time requirements, while in the IT field, high throughput and affordable equipment are generally demanded. Although Ethernet and IEEE 802.11 cover different scenarios, their popularity and availability as COTS solutions in IT has not escaped the attention of OT network designers. This has triggered a trend where OT and IT are converging to create solutions that attempt to take advantage of both worlds. One example of this is TSN, a wired solution for OT based on Ethernet, which is becoming increasingly popular. Unfortunately, the technologies in the wireless area are not yet as advanced as TSN in the wired area. A wireless counterpart to TSN based on IEEE 802.11 would be beneficial to take advantage of low-cost, high-throughput COTS devices while still providing a solution with mobility. The problem is that IEEE 802.11 in its current form does not support dependable real-time applications. The reasons are the coordination between network nodes that IEEE 802.11 uses when accessing the channel and the problems of the wireless channel such as path loss, shadowing and multipath fading.

10.1 Revisiting the research hypothesis

In view of the above problem, the research work presented in this thesis began with the formulation of the hypothesis:

High-throughput wireless COTS based on IEEE 802.11 can support reliable real-time communications with sufficient quality to be integrated into existing reliable real-time wired networks based on Ethernet.

The currently available high-throughput wireless COTS devices based on IEEE 802.11 do not support reliable real-time communications. Recognizing how different requirements are achieved at the expense of others, Contribution area 1 (Chapter 7) presents an analysis of the trade-offs between the typical requirements of OT and IT. This analysis was carried out with the intention of shedding light on what requirements can be requested while enabling the development of realistic solutions. Based on this analysis, a solution framework was proposed consisting of a wireless network relying on IEEE 802.11 COTS hardware. The wireless network is designed to complement a wired Ethernet-based network using TTE or TSN technologies to meet dependability and real-time requirements. To address the channel access coordination difficulties in IEEE 802.11, a TDMA MAC protocol is proposed. The integration between the wireless and wired network relies on the existing technology based on IEEE 802.1 that supports interoperability between IEEE 802.11 and Ethernet. This is complemented by a scheduling solution that enables data exchanges with different occurrence patterns and criticality, while meeting delivery latency and jitter, as well as delivery deadline requirements. Contribution area 1 also serves to identify the overall problems, i.e., faults, that such a wireless solution must address to ensure the fulfillment of dependable and real-time requirements with sufficient quality to be integrated into existing reliable real-time wired networks based on Ethernet. The remaining contributions deal with the resolution of these faults. Contribution area 2 (Chapter 8) proposes several mechanisms for fault prevention, including cognitive radio, and Contribution area 3 (Chapter 9) proposes several mechanisms for fault tolerance, including time and space redundancy.

The proposed mechanisms were evaluated using various techniques, including reliability analysis, computer simulations and a hardware implementation. To investigate the impact of faults on the system, several challenging scenarios were created in which the performance of the system in coping with these faults was measured. The results show that the proposed solutions are able to provide reliable real-time wireless communications at a level that they can interoperate with reliable real-time wired networks, which proves the hypothesis. However, the results also show that the degree of fulfillment of the reliability and real-time requirements depends on the scenario.

10.2 Revisiting the research questions

The list of research questions formulated as a guide for the research work is reviewed below together with the contributions that address them.

RQ1 – On the requirements to be addressed: *What are the trade-offs between the targeted solution requirements, in particular dependability and real-time requirements, for data communication systems with integrated wired and wireless connectivity?*

Data communication scenarios in industrial automation, automotive, avionics and robotics are associated with demanding requirements. A comprehensive overview of the typical requirements for such scenarios is given in Chapter 3. The overview of communication technologies in Chapter 5 illustrates how difficult it is to cover the various requirements with a single solution, as compromises have to be made and the technologies can be quite specialized after several decades of evolving deployments. The review highlights the gap between the dependability and real-time guarantees of OT technologies on the one hand and the high throughput and lower costs of IT technologies on the other. Similarly, wired technologies are often the preferred choice when it comes to dependability and real-time requirements, sacrificing wireless networks and their potential benefits, such as mobility. Taking these considerations into account, Contribution area 1 (Chapter 7) provides an analysis of the trade-offs between the requirements, which serves as the first step towards outlining a solution.

RQ2 – On the faults to be addressed: *How to overcome the likely faults in systems with integrated wired and wireless connectivity so that the targeted requirements, in particular reliability and real-time requirements, are still fulfilled?*

Contribution area 1 (Chapter 7) describes and analyzes three relevant faults that can occur in a communication system that targets dependability and real-time requirements, while integrating wired and wireless connectivity. In this integrated wired and wireless system, the wired segment is based on Ethernet deployed together with TTE or TSN, and the wireless segment is based on IEEE 802.11.

Fault 1 refers to a delayed transmission due to the lack of coordination between data senders. This fault arises because of DCF, the basic MAC protocol in IEEE 802.11, since it is based on sensing the channel before attempting a transmission. Considering that the medium in wireless settings is shared by multiple nodes that are in the same broadcast domain, this behavior could cause a node to sense the medium as busy indefinitely as long as other nodes keep transmitting data. As a result, the nodes' transmissions are delayed while they wait for an opportunity to transmit.

Fault 2 describes a lost transmission due to path loss, shadowing, multipath fading, overlapping transmissions and/or interference. The problems with overlapping transmissions and interference can also be partially attributed to the DCF MAC, which in certain situations could cause two or more nodes to sense the medium as free and start a transmission at the same time. Interference could also be caused by other devices transmitting simultaneously on the same channel and beyond the control of the network. Fault 2 could also be caused by typical wireless channel problems such as path loss, shadowing and multipath fading.

Fault 3 focuses on the wired network problems and represents a lost transmission due to broken wires.

RQ2.1: *What fault-prevention mechanisms can be used to support reliable and real-time communications for the wireless medium?*

The fault-prevention mechanisms are presented in Contribution area 2 (Chapter 8).

Fault 1 and Fault 2 are addressed by the proposal of `MAC_CH`, a TDMA MAC protocol that coordinates the access to the medium and enables differentiated handling according to timing and criticality requirements. The differentiated treatment is based on the support of TTE traffic classes or various TSN mechanisms. Furthermore, the mechanisms `MAC_BH_1`, `MAC_BH_2` and `MAC_BH_3` for BE traffic are proposed to overcome the challenges of using a shared half-duplex medium when transmitting over IEEE 802.11. A scheduling solution is presented that extends the scheduler for TTE and TSN that is already available for the wired segments. The scheduling solution enables data transmissions that span the wired and wireless segments of the network, while considering their timing and criticality requirements. In this way, unwanted delays and overlaps in transmission caused by faults 1 and 2 can be avoided.

Fault 2 is also handled by a frequency diversity mechanism based on cognitive radio `MAC_CR`. The approach increases the probability of successful transmission by selecting the frequency at which the probability of interference is lowest. The scheduling solution introduced for MAC_CH is extended to include cognitive radio.

The evaluation of MAC_CH is carried out via a hardware implementation. The implementation is based on IEEE 802.11 COTS hardware. The results show that admission control is crucial to avoid a congested medium leading to delayed or overlapping transmissions and that the proposed mechanism is able to provide more constrained jitter values. The mechanisms MAC_BH_1, MAC_BH_2 and MAC_BH_3 are evaluated by simulation. The simulation results clearly show that the choice of each of the three proposed mechanisms depends on the traffic pattern. The more is known about the traffic, the better is the performance improvement that can be achieved by selecting the appropriate mechanism. MAC_CR is evaluated by simulation and the results show that the cognitive radio approach greatly reduces the occurrence of faults due to interference.

RQ2.2: *What fault-tolerance mechanisms can be used to support reliable and real-time communications for systems with integrated wired and wireless connectivity?*

The fault-tolerance mechanisms are presented in Contribution area 3 (Chapter 9).

Fault 1 and Fault 2 are also handled by the proposal of MAC_TD_Cons and MAC_TD-_Sprd, which offer redundancy in the time dimension through retransmissions. The scheduling solution introduced for MAC_CH is extended to include retransmissions. Two variants of MAC mechanisms are investigated, one version that utilizes all scheduled retransmissions and another version that is based on the use of feedback about the retransmission success and allows the reuse of the unnecessary retransmission slots. Two reliability mechanisms are also applied to these variants, with retransmissions occurring either immediately, MAC_TD_Cons, or slightly later, MAC_TD_Sprd.

The performance of MAC_TD_Cons and MAC_TD_Sprd is evaluated using simulations. The results show that retransmissions are an effective means to increase the reliability of the communication system and that the different retransmission schemes can be applied to better cope with interference of different intensity, duration and persistence. A comparison is also made between MAC_CR, MAC_TD_Cons and MAC_TD_Sprd. The simulation results show that the MAC_CR scheme performs better than MAC_TD-_Cons and MAC_TD_Sprd in all environments considered. Moreover, the performance improvement does not come at the expense of an increase in data delivery delays, as is the case with MAC_TD_Cons and MAC_TD_Sprd, but as a drawback MAC_CR requires specialized hardware.

Fault 3 is addressed with redundancy in the space dimension, with the proposal of the mechanisms MAC_SD_Sta and MAC_SD_Dyn. The mechanisms utilize a wireless backup network to provide an alternate data path for transmissions originally scheduled over the wired network. One redundancy mechanism, MAC_SD_Sta, statically assigns communication slots in the wireless segment and operates in parallel with the wired network. The other mechanism, MAC_SD_Dyn, reacts dynamically to broken wired links and establishes an alternative wireless connection.

The reliability when using MAC_SD_Sta or MAC_SD_Dyn is evaluated by reliability analysis and by simulation, with both methods showing similar behavior. The results show that using a wireless backup network as a fault tolerance mechanism can effectively increase the reliability of a wired network by avoiding the problem of permanent faults due to broken wires. The choice of one of the proposed redundancy mechanisms is subject to a trade-off between ensuring the most critical data messages in the case of `MAC_SD_Sta` or better average performance in the case of `MAC_SD_Dyn`.

The fault prevention and fault tolerance mechanisms proposed in this thesis, which utilize the diversity of communication channels and put the focus on scheduling the communication resources, show how reliability and real-time guarantees can be met in critical scenarios using COTS hardware based on IEEE 802.11 and enable the integration into existing reliable real-time networks based on Ethernet.

10.3 Future work

There are several research directions not covered in this thesis that would be relevant to explore. One of them is the evaluation of the presented mechanisms in more diverse and realistic scenarios, such as real-world deployments in industrial automation. This would help to tailor a solution to the specifics of each environment, e.g., by selecting the subset of the mechanisms proposed in this thesis that provide the most benefit. Further, extending the hardware implementation to support the mechanisms not yet implemented, i.e., cognitive radio, time diversity and space diversity, would help to uncover any challenges of the proposed mechanisms when porting them to the hardware. Similarly, the implementation could be extended to create a hybrid wired/wireless network with TSN and Ethernet on the wired segments and discover any issues of the integrated wired/wireless approach on the hardware. Another topic worth exploring is the clock synchronization mechanisms in wireless deployments, which could aim for better synchronization precision in the presented hardware implementation. Finally, it is worth exploring the combination with additional fault-tolerance and fault-prevention mechanisms for the wireless medium based on standardized solutions. For example, the spatial diversity standardized in IEEE 802.11 can help to increase the reliability of the wireless medium by performing parallel transmissions in the so-called spatial streams.

References

[1] Pablo Gutiérrez Peón, Hermann Kopetz, and Wilfried Steiner. Towards a Reliable and High-Speed Wireless Complement to TTEthernet. In *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Barcelona, Spain, 2014.

[2] Pablo Gutiérrez Peón, Elisabeth Uhlemann, Wilfried Steiner, and Mats Björkman. A Wireless MAC Method with Support for Heterogeneous Data Traffic. In *Proceedings of the Annual Conference of the IEEE Industrial Electronics Society (IECON)*, pages 3869–3874, Yokohama, Japan, 2015.

[3] Pablo Gutiérrez Peón, Elisabeth Uhlemann, Wilfried Steiner, and Mats Björkman. Medium Access Control for Wireless Networks with Diverse Time and Safety Real-Time Requirements. In *Proceedings of the Annual Conference of the IEEE Industrial Electronics Society (IECON)*, pages 4665–4670, Florence, Italy, 2016.

[4] Pablo Gutiérrez Peón, Elisabeth Uhlemann, Wilfried Steiner, and Mats Bjorkman. Applying Time Diversity for Improved Reliability in a Real-Time Heterogeneous MAC Protocol. In *Proceedings of the IEEE Vehicular Technology Conference (VTC-Spring)*, Sydney, Australia, 2017.

[5] Pablo Gutiérrez Peón, Pedro Manuel Rodríguez, Zaloa Fernández, Francisco Pozo, Elisabeth Uhlemann, Iñaki Val, and Wilfried Steiner. Cognitive Radio for Improved Reliability in a Real-Time Wireless MAC Protocol based on TDMA. In *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Limassol, Cyprus, 2017.

[6] Pablo Gutiérrez Peón, Wilfried Steiner, and Elisabeth Uhlemann. Network Fault Tolerance by Means of Diverse Physical Layers. In *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1697–1704, Vienna, Austria, 2020.

[7] Pablo Gutiérrez Peón, Paraskevas Karachatzis, Wilfried Steiner, and Elisabeth Uhlemann. Time-Sensitive Networking's Scheduled Traffic Implementation on IEEE 802.11 COTS Devices. In *Proceedings of the International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, pages 167–175, Niigata, Japan, 2023.

[8] ISO/IEC 7498 - Information technology - Open Systems Interconnection - Basic Reference Model: The Basic Model. Standard, International Organization for

Standardization (ISO) and International Electrotechnical Commission (IEC), Geneva, Switzerland, 1994.

[9] Algirdas Avizienis, J. C. Laprie, Brian Randell, and Carl Landwehr. Basic Concepts and Taxonomy of Dependable and Secure Computing. *IEEE Transactions on Dependable and Secure Computing*, 1(1):11–33, 2004.

[10] Barry W. Johnson. *Design & Analysis of Fault Tolerant Digital Systems.* Addison-Wesley Longman Publishing Co., Inc., Boston, USA, 1988.

[11] Wilfried Steiner. An Evaluation of SMT-based Schedule Synthesis For Time-Triggered Multi-Hop Networks. In *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, pages 375–384, San Diego, USA, 2010.

[12] IEEE 802.3-2018 - IEEE Standard for Ethernet. Standard, Institute of Electrical and Electronics Engineers (IEEE), New York, USA, 2018.

[13] IEEE 802.11-2020 - IEEE Standard for Information Technology – Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks – Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Standard, Institute of Electrical and Electronics Engineers (IEEE), New York, USA, 2021.

[14] Andreas Willig, Kirsten Matheus, and Adam Wolisz. Wireless Technology in Industrial Networks. *Proceedings of the IEEE*, 93(6):1130–1151, 2005.

[15] Christian M. Fuchs. The Evolution of Avionics Networks From ARINC 429 to AFDX. *Innovative Internet Technologies and Mobile Communications (IITM), and Aerospace Networks (AN)*, 65:1551–3203, 2012.

[16] Thomas Nolte, Hans Hansson, and Lucia Lo Bello. Automotive Communications - Past, Current and Future. In *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Catania, Italy, 2005.

[17] Luis Almeida, Frederico Santos, Tullio Facchinetti, Paulo Pedreiras, Valter Silva, and L. Seabra Lopes. Coordinating distributed autonomous agents with a real-time database: The CAMBADA project. In *Proceedings of the International Symposium on Computer and Information Sciences (ISCIS)*, pages 876–886, Kemer-Antalya, Turkey, 2004.

[18] IEC 61158 - Industrial communication networks - Fieldbus specifications - Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series. Standard, International Electrotechnical Commission (IEC), Geneva, Switzerland, 2019.

[19] ISO 11898-1 - Road vehicles - Controller area network (CAN) - Part 1: Data link layer and physical signalling. Standard, International Organization for Standardization (ISO), Geneva, Switzerland, 2015.

[20] Mohammad Ashjaei, Lucia Lo Bello, Masoud Daneshtalab, Gaetano Patti, Sergio Saponara, and Saad Mubeen. Time-Sensitive Networking in automotive embedded systems: State of the art and research opportunities. *Journal of Systems Architecture*, 117:102137, 2021.

[21] Wilfried Steiner, Pablo Gutiérrez Peón, Marina Gutiérrez, Ayhan Mehmed, Guillermo Rodriguez-Navas, Elena Lisova, and Francisco Pozo. Next Generation Real-Time Networks Based on IT Technologies. In *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Berlin, Germany, 2016.

[22] IEC 61784-3 - Industrial communication networks - Profiles - Part 3: Functional safety fieldbuses - General rules and profile definitions. Standard, International Electrotechnical Commission (IEC), Geneva, Switzerland, 2021.

[23] ARINC Specification 664 - Aircraft Data Network, Part 7 - Avionics Full Duplex Switched Ethernet (AFDX). Standard, Aeronautical Radio Inc, Annapolis, USA, 2005.

[24] AS6802 - Time-Triggered Ethernet. Standard, SAE International, Warrendale, USA, 2009.

[25] IEEE 802.1AS-2020 - IEEE Standard for Local and Metropolitan Area Networks – Timing and Synchronization for Time-Sensitive Applications. Standard, Institute of Electrical and Electronics Engineers (IEEE), New York, USA, 2020.

[26] IEEE 802.1Q-2018 - IEEE Standard for Local and Metropolitan Area Network – Bridges and Bridged Networks. Standard, Institute of Electrical and Electronics Engineers (IEEE), New York, USA, 2018.

[27] IEEE 802.1CB-2017 - IEEE Standard for Local and metropolitan area networks – Frame Replication and Elimination for Reliability. Standard, Institute of Electrical and Electronics Engineers (IEEE), New York, USA, 2017.

[28] Luis Silva, Paulo Pedreiras, Pedro Fonseca, and Luis Almeida. On the adequacy of SDN and TSN for Industry 4.0. In *Proceedings of the International Symposium on Real-Time Distributed Computing (ISORC)*, pages 43–51, Valencia, Spain, 2019.

[29] Tommaso Fedullo, Alberto Morato, Federico Tramarin, Luigi Rovati, and Stefano Vitturi. A Comprehensive Review on Time Sensitive Networks with a Special Focus on Its Applicability to Industrial Smart and Distributed Measurement Systems. *MDPI Sensors*, 22(4):1638, 2022.

[30] Michael Johas Teener, Allen Huotari, Yongbum Kim, Rick Kreifeldt, and Kevin Stanton. No-excuses Audio/Video Networking: the Technology Behind AVnu. White paper, AVnu Alliance, Beaverton, United States, 2009.

[31] Carla-Fabiana Chiasserini and Ramesh R. Rao. Coexistence Mechanisms for Interference Mitigation in the 2.4-GHz ISM Band. *IEEE Transactions on Wireless Communications*, 2(5):964–975, 2003.

[32] Roel Snieder and Ken Larner. *The Art of Being a Scientist: A Guide for Graduate Students*. Cambridge University Press, Cambridge, United Kingdom, 2009.

[33] Andreas F. Molisch. *Wireless Communications*. John Wiley & Sons, Chichester, United Kingdom, 2012.

[34] Pedro Manuel Rodríguez, Raúl Torrego, Félix Casado, Zaloa Fernández, Mikel Mendicute, Aitor Arriola, and Iñaki Val. Dynamic Spectrum Access Integrated in a Wideband Cognitive RF-Ethernet Bridge for Industrial Control Applications. *Journal of Signal Processing Systems*, 83(1):19–28, 2016.

[35] Lucia Lo Bello and Emanuele Toscano. Coexistence Issues of Multiple Co-Located IEEE 802.15.4/ZigBee Networks Running on Adjacent Radio Channels in Industrial Environments. *IEEE Transactions on Industrial Informatics*, 5(2):157–167, 2009.

[36] Ilenia Tinnirello, Daniele Croce, Natale Galioto, Domenico Garlisi, and Fabrizio Giuliano. Cross-Technology WiFi/ZigBee Communications: Dealing With Channel Insertions and Deletions. *IEEE Communications Letters*, 20(11):2300–2303, 2016.

[37] Alan Burns and Robert Davis. Mixed Criticality Systems - A Review. Technical report, Department of Computer Science, University of York, York, United Kingdom, 2013.

[38] Francisco Pozo, Guillermo Rodriguez-Navas, and Hans Hansson. Methods for Large-Scale Time-Triggered Network Scheduling. *Multidisciplinary Digital Publishing Institute Electronics*, 8(7):738, 2019.

[39] RTCA/DO-178B - Software Considerations in Airborne Systems and Equipment Certification. Standard, RTCA/DO, Washington DC, USA, 1992.

[40] Robson Costa, Paulo Portugal, Francisco Vasques, Carlos Montez, and Ricardo Moraes. Limitations of the IEEE 802.11 DCF, PCF, EDCA and HCCA to handle real-time traffic. In *Proceedings of the IEEE International Conference on Industrial Informatics (INDIN)*, pages 931–936, Cambridge, United Kingdom, 2015.

[41] C. Casetti, C.-F. Chiasserini, M. Fiore, and M. Garetto. Notes on the Inefficiency of 802.11e HCCA. In *Proceedings of the IEEE Vehicular Technology Conference (VTC-Fall)*, pages 2513–2517, Dallas, USA, 2005.

[42] Jose R. Betiol Junior, Jim Lau, Luciana de Oliveira Rech, Analcia Schiaffino Morales, and Ricardo Moraes. Experimental Evaluation of the Coexistence of IEEE 802.11 EDCA and DCF Mechanisms. In *Proceedings of the IEEE Symposium on Computers and Communications (ISCC)*, pages 847–852, Natal, Brazil, 2018.

[43] Lifei Huang and Ten-Hwang Lai. On the Scalability of IEEE 802.11 Ad Hoc Networks. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 173–182, Lausanne, Switzerland, 2002.

[44] Robert I. Davis, Alan Burns, Reinder J. Bril, and Johan J. Lukkien. Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised. *Real-Time Systems*, 35(3):239–272, 2007.

[45] ISO 17458-1 - Road vehicles - FlexRay communications system - Part 1: General information and use case definition. Standard, International Organization for Standardization (ISO), Geneva, Switzerland, 2013.

[46] Traian Pop, Paul Pop, Petru Eles, Zebo Peng, and Alexandru Andrei. Timing analysis of the FlexRay communication protocol. *Real-time systems*, 39:205–235, 2008.

[47] J. Javier Gutiérrez, J. Carlos Palencia, and Michael González Harbour. Response time analysis in AFDX networks. In *Proceedings of the Jornadas de Tiempo Real*, Madrid, Spain, 2011.

[48] Eduardo Tovar and Francisco Vasques. Real-Time Fieldbus Communications Using Profibus Networks. *IEEE Transactions on Industrial Electronics*, 46(6):1241–1251, 1999.

[49] IEEE 1588-2019 - IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. Standard, Institute of Electrical and Electronics Engineers (IEEE), New York, USA, 2019.

[50] Gunnar Prytz. A performance analysis of EtherCAT and PROFINET IRT. In *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 408–415, Hamburg, Germany, 2008.

[51] Gianluca Cena, Lucia Seno, Adriano Valenzano, and Stefano Vitturi. Performance analysis of Ethernet Powerlink networks for distributed control and automation systems. *Computer Standards & Interfaces*, 31(3):566–572, 2009.

[52] IEC 61784-2 - Industrial communication networks - Profiles - Part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC/IEEE 8802-3. Standard, International Electrotechnical Commission (IEC), Geneva, Switzerland, 2019.

[53] IEC 61800-7 - Adjustable speed electrical power drive systems - Part 7-1: Generic interface and use of profiles for power drive systems - Interface definition. Standard, International Electrotechnical Commission (IEC), Geneva, Switzerland, 2015.

[54] IEC 62439-3 - Industrial communication networks - High availability automation networks - Part 3: Parallel Redundancy Protocol (PRP) and High-availability Seamless Redundancy (HSR). Standard, International Electrotechnical Commission (IEC), Geneva, Switzerland, 2021.

[55] IEC 62439-2 - Industrial communication networks - High availability automation networks - Part 2: Media Redundancy Protocol (MRP). Standard, International Electrotechnical Commission (IEC), Geneva, Switzerland, 2021.

[56] Paulo Pedreiras, Luis Almeida, and Paolo Gai. The FTT-Ethernet protocol: Merging flexibility, timeliness and efficiency. In *Proceedings of the IEEE Euromicro Conference on Real-Time Systems (ECRTS)*, pages 152–152, Vienna, Austria, 2002.

[57] Silviu S. Craciunas, Ramon Serna Oliver, Martin Chmelík, and Wilfried Steiner. Scheduling Real-Time Communication in IEEE 802.1Qbv Time Sensitive Networks. In *Proceedings of the ACM International Conference on Real-Time Networks and Systems (RTNS)*, pages 183–192, Brest, France, 2016.

[58] Ahmed Nasrallah, Akhilesh S Thyagaturu, Ziyad Alharbi, Cuixiang Wang, Xing Shao, Martin Reisslein, and Hesham Elbakoury. Performance Comparison of IEEE 802.1 TSN Time Aware Shaper (TAS) and Asynchronous Traffic Shaper (ATS). *IEEE Access*, 7:44165–44181, 2019.

[59] IEEE 802.15.1-2005 - IEEE Standard for Information technology – Local and metropolitan area networks – Specific requirements– Part 15.1a: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Wireless Personal Area Networks (WPAN). Standard, Institute of Electrical and Electronics Engineers (IEEE), New York, USA, 2005.

[60] Raúl Rondón, Mikael Gidlund, and Krister Landernäs. Evaluating Bluetooth Low Energy Suitability for Time-Critical Industrial IoT Applications. *International Journal of Wireless Information Networks*, 24(3):278–290, 2017.

[61] Jimmy Kjellsson, Anne Elisabeth Vallestad, Richard Steigmann, and Dacfey Dzung. Integration of a Wireless I/O Interface for PROFIBUS and PROFINET for Factory Automation. *IEEE Transactions on Industrial Electronics*, 56(10):4279–4287, 2009.

[62] IEEE 802.15.4-2020 - IEEE Standard for Low-Rate Wireless Networks. Standard, Institute of Electrical and Electronics Engineers (IEEE), New York, USA, 2020.

[63] IEC 62591 - Industrial networks - Wireless communication network and communication profiles - WirelessHART. Standard, International Electrotechnical Commission (IEC), Geneva, Switzerland, 2016.

[64] Abusayeed Saifullah, You Xu, Chenyang Lu, and Yixin Chen. End-to-End Delay Analysis for Fixed Priority Scheduling in WirelessHART Networks. In *Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 13–22, Chicago, USA, 2011.

[65] Junyang Shi, Mo Sha, and Zhicheng Yang. Distributed Graph Routing and Scheduling for Industrial Wireless Sensor-Actuator Networks. *IEEE/ACM Transactions on Networking*, 27(4):1669–1682, 2019.

[66] Sihoon Moon, Hyungseok Park, Hoon Sung Chwa, and Kyung-Joon Park. AdaptiveHART: An Adaptive Real-Time MAC Protocol for Industrial Internet-of-Things. *IEEE Systems Journal*, 16(3):4849–4860, 2022.

[67] ANSI/ISA-100.11a-2011 Wireless systems for industrial automation: Process control and related applications. Standard, International Society of Automation (ISA), Durham, USA, 2009.

[68] Favian Dewanta, Fadillah Purnama Rezha, and Dong-Sung Kim. Message scheduling approach on dedicated time slot of ISA100.11a. In *Proceedings of the International Conference on Information and Communication Technology Convergence (ICTC)*, pages 466–471, Jeju Island, Korea, 2012.

[69] Xavier Vilajosana, Thomas Watteyne, Tengfei Chang, Mališa Vučinić, Simon Duquennoy, and Pascal Thubert. IETF 6TiSCH: A Tutorial. *IEEE Communications Surveys & Tutorials*, 22(1):595–615, 2019.

[70] Nicola Accettura, Elvis Vogli, Maria Rita Palattella, Luigi Alfredo Grieco, Gennaro Boggia, and Mischa Dohler. Decentralized Traffic Aware Scheduling in 6TiSCH Networks: Design and Experimental Evaluation. *IEEE Internet of Things Journal*, 2(6):455–470, 2015.

[71] Zigbee document 053474r06 version 1.0. Standard, Zigbee Alliance Std, Davis, USA, 2004.

[72] Industrial networks - Wireless communication network and communication profiles - WIA-FA. Standard, International Electrotechnical Commission (IEC), Geneva, Switzerland, 2017.

[73] Alexander Hellemans. Manufacturing Mayday. *IEEE Spectrum*, 44(1):10–13, 2007.

[74] Armin Hadziaganović, Mahin K. Atiq, Thomas Blazek, Hans-Peter Bernhard, and Andreas Springer. The performance of openSAFETY protocol via IEEE 802.11 wireless communication. In *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Västerås, Sweden, 2021.

[75] Ricardo Moraes, Francisco Vasques, Paulo Portugal, and José Alberto Fonseca. VTP-CSMA: A Virtual Token Passing Approach for Real-Time Communication in IEEE 802.11 Wireless Networks. *IEEE Transactions on Industrial Informatics*, 3(3):215–224, 2007.

[76] Jungbo Son, Hosuk Choi, and Sin-Chong Park. An effective polling MAC scheme for IEEE 802.11e. In *Proceedings of the IEEE International Symposium on Communications and Information Technologies (ISCIT)*, pages 296–301, Sapporo, Japan, 2004.

[77] Claudio Cicconetti, Luciano Lenzini, Enzo Mingozzi, and Giovanni Stea. Design and performance analysis of the Real-Time HCCA scheduler for IEEE 802.11e WLANs. *Computer Networks*, 51(9):2311–2325, 2007.

[78] Tullio Facchinetti, Luis Almeida, Giorgio C. Buttazzo, and Carlo Marchini. Real-Time Resource Reservation Protocol for Wireless Mobile Ad Hoc Networks. In *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, pages 382–391, Lisbon, Portugal, 2004.

[79] Magnus Jonsson and Kristina Kunert. MC-EDF: A Control-Channel based Wireless Multichannel MAC Protocol with Real-Time Support. In *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Kraków, Poland, 2012.

[80] Lucia Seno, Gianluca Cena, Adriano Valenzano, and Claudio Zunino. Bandwidth Management for Soft Real-Time Control Applications in Industrial Wireless Networks. *IEEE Transactions on Industrial Informatics*, 13(5):2484–2495, 2017.

[81] Henning Trsek and Jürgen Jasperneite. An isochronous medium access for real-time wireless communications in industrial automation systems - A use case for wireless clock synchronization. In *Proceedings of the International IEEE Symposium on Precision Clock Synchronization for Measurement, Control and Communication (ISPCS)*, pages 81–86, Munich, Germany, 2011.

[82] Zaloa Fernandez, Oscar Seijo, Mikel Mendicute, and Iñaki Val. Analysis and Evaluation of a Wired/Wireless Hybrid Architecture for Distributed Control Systems With Mobility Requirements. *IEEE Access*, 7:95915–95931, 2019.

[83] Óscar Seijo, Jesús Alberto López-Fernández, and Iñaki Val. w-SHARP: Implementation of a High-Performance Wireless Time-Sensitive Network for Low Latency and Ultra-Low Cycle Time Industrial Applications. *IEEE Transactions on Industrial Informatics*, 17(5):3651–3662, 2020.

[84] Yi-Hung Wei, Quan Leng, Song Han, Aloysius K. Mok, Wenlong Zhang, and Masayoshi Tomizuka. RT-WiFi: Real-Time High-Speed Communication Protocol for Wireless Cyber-Physical Control Applications. In *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, pages 140–149, Vancouver, Canada, 2013.

[85] Robson Costa, Paulo Portugal, Francisco Vasques, and Ricardo Moraes. A TDMA-based Mechanism for Real-Time Communication in IEEE 802.11e Networks. In *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Bilbao, Spain, 2010.

[86] Robson Costa, Jim Lau, Paulo Portugal, Francisco Vasques, and Ricardo Moraes. Handling real-time communication in infrastructured IEEE 802.11 wireless networks: The RT-WiFi approach. *Journal of Communications and Networks*, 21(3):319–334, 2019.

[87] Frederico Santos, Luís Almeida, and Luís Seabra Lopes. Self-configuration of an adaptive TDMA wireless communication protocol for teams of mobile robots. In *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1197–1204, Hamburg, Germany, 2008.

[88] Alberto Morato, Stefano Vitturi, Federico Tramarin, Claudio Zunino, and Manuel Cheminod. Time-Sensitive Networking to Improve the Performance of Distributed Functional Safety Systems Implemented over Wi-Fi. *MDPI Sensors*, 23(18):7825, 2023.

[89] Jungsook Kim, Jaehan Lim, Christopher Pelczar, and Byungtae Jang. RRMAC: A sensor network MAC for real time and reliable packet transmission. In *Proceedings of the IEEE International Symposium on Consumer Electronics (ISCE)*, Vilamoura-Algarve, Portugal, 2008.

[90] Wei Shen, Tingting Zhang, Mikael Gidlund, and Felix Dobslaw. SAS-TDMA: A source aware scheduling algorithm for real-time communication in industrial wireless sensor networks. *Wireless networks*, 19:1155–1170, 2013.

[91] Hossam Farag, Mikael Gidlund, and Patrik Österberg. A Delay-Bounded MAC Protocol for Mission- and Time-Critical Applications in Industrial Wireless Sensor Networks. *IEEE Sensors Journal*, 18(6):2607–2616, 2018.

[92] J. A. Afonso, L. A. Rocha, H. R. Silva, and J. H. Correia. MAC Protocol for Low-Power Real-Time Wireless Sensing and Actuation. In *Proceedings of the IEEE International Conference on Electronics, Circuits, and Systems (ICECS)*, pages 1248–1251, Nice, France, 2006.

[93] Federico Terraneo, Paolo Polidori, Alberto Leva, and William Fornaciari. TDMH-MAC: Real-time and multi-hop in the same wireless MAC. In *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, pages 277–287, Nashville, USA, 2018.

[94] Thomas Watteyne, Isabelle Augé-Blum, and Stéphane Ubéda. Dual-Mode Real-Time MAC protocol for Wireless Sensor Networks: a Validation/Simulation Approach. In *Proceedings of the ACM Integrated Internet Ad Hoc and Sensor Networks (InterSense)*, Nice, France, 2006.

[95] Tao Zheng, Mikael Gidlund, and Johan Åkerberg. WirArb: A New MAC Protocol for Time Critical Industrial Wireless Sensor Network Applications. *IEEE Sensors Journal*, 16(7):2127–2139, 2015.

[96] Paulo Bartolomeu, Muhammad Alam, Joaquim Ferreira, and José Alberto Fonseca. Supporting Deterministic Wireless Communications in Industrial IoT. *IEEE Transactions on Industrial Informatics*, 14(9):4045–4054, 2018.

[97] Zaloa Fernandez, Pedro Manuel Rodríguez, Iñaki Val, and Mikel Mendicute. An improved wireless MAC protocol for priority based data delivery. In *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Berlin, Germany, 2016.

[98] Lucia Seno, Gianluca Cena, Stefano Scanzio, Adriano Valenzano, and Claudio Zunino. Enhancing Communication Determinism in Wi-Fi Networks for Soft Real-Time Industrial Applications. *IEEE Transactions on Industrial Informatics*, 13(2):866–876, 2016.

[99] Duc Khai Lam, Keishi Yamaguchi, Yasuhiro Shinozaki, Satoshi Morita, Yuhei Nagao, Masayuki Kurosaki, and Hiroshi Ochi. A fast industrial WLAN protocol and its MAC implementation for factory communication systems. In *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Luxemburg, 2015.

[100] Paulo Bartolomeu, José Fonseca, and Francisco Vasques. Implementing the wireless FTT protocol: A feasibility analysis. In *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Bilbao, Spain, 2010.

[101] Henning Trsek, Stefan Schwalowsky, Bjoern Czybik, and Juergen Jasperneite. Implementation of an advanced IEEE 802.11 WLAN AP for real-time wireless communications. In *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Toulouse, France, 2011.

[102] Carl Lusty, Vladimir Estivill-Castro, and René Hexel. TTWiFi: Time-Triggered Communication over WiFi. In *Proceedings of the ACM Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications (DIVANet)*, pages 35–44, Alicante, Spain, 2021.

[103] Gianluca Cena, Stefano Scanzio, and Adriano Valenzano. SDMAC: A Software-Defined MAC for Wi-Fi to Ease Implementation of Soft Real-Time Applications. *IEEE Transactions on Industrial Informatics*, 15(6):3143–3154, 2018.

[104] Zelin Yun, Peng Wu, Shengli Zhou, Aloysius K. Mok, Mark Nixon, and Song Han. RT-WiFi on Software-Defined Radio: Design and Implementation. In *Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 254–266, Milan, Italy, 2022.

[105] Yujun Cheng, Dong Yang, and Huachun Zhou. Det-WiFi: A multihop TDMA MAC implementation for industrial deterministic applications based on commodity 802.11 hardware. *Wireless Communications and Mobile Computing*, 2017, 2017.

[106] Vishal Sevani, Bhaskaran Raman, and Piyush Joshi. Implementation-Based Evaluation of a Full-Fledged Multihop TDMA-MAC for WiFi Mesh Networks. *IEEE Transactions on Mobile Computing*, 13(2):392–406, 2012.

[107] Óscar Seijo, Xabier Iturbe, and Iñaki Val. Tackling the Challenges of the Integration of Wired and Wireless TSN with a Technology Proof-of-Concept. *IEEE Transactions on Industrial Informatics*, 18(10):7361–7372, 2021.

[108] Ben Schneider, Rute C. Sofia, and Matthias Kovatsch. A Proposal for Time-Aware Scheduling in Wireless Industrial IoT Environments. In *Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS)*, Budapest, Hungary, 2022.

[109] Susruth Sudhakaran, Vincent Mageshkumar, Amit Baxi, and Dave Cavalcanti. Enabling QoS for Collaborative Robotics Applications with Wireless TSN. In *Proceedings of the IEEE International Conference on Communications Workshops (ICC Workshops)*, Montreal, Canada, 2021.

[110] Gennaro Boggia, Pietro Camarda, Luigi Alfredo Grieco, and Giammarco Zacheo. An experimental evaluation on using TDMA over 802.11 MAC for Wireless Networked Control Systems. In *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1157–1160, Hamburg, Germany, 2008.

[111] Wim Torfs and Chris Blondia. TDMA on commercial of-the-shelf hardware: Fact and fiction revealed. *AEU - International Journal of Electronics and Communications*, 69(5):800–813, 2015.

[112] Aneeq Mahmood, Reinhard Exel, Henning Trsek, and Thilo Sauter. Clock Synchronization Over IEEE 802.11 - A Survey of Methodologies and Protocols. *IEEE Transactions on Industrial Informatics*, 13(2):907–922, 2016.

[113] Aneeq Mahmood, Georg Gaderer, Henning Trsek, Stefan Schwalowsky, and Nikolaus Kerö. Towards High Accuracy in IEEE 802.11 based Clock Synchronization using PTP. In *Proceedings of the IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication (ISPCS)*, pages 13–18, Munich, Germany, 2011.

[114] Iñaki Val, Óscar Seijo, Raul Torrego, and Armando Astarloa. IEEE 802.1AS Clock Synchronization Performance Evaluation of an Integrated Wired-Wireless TSN Architecture. *IEEE Transactions on Industrial Informatics*, 18(5):2986–2999, 2021.

[115] A. Mishra, T. Kim, and G. Masson. CH-MAC: A multi-channel MAC protocol for dynamic spectrum access networks. In *Proceedings of the International Symposium on Wireless Personal Multimedia Communications (WPMC)*, Atlantic City, USA, 2013.

[116] José Marinho and Edmundo Monteiro. Enhanced protection of hidden primary users through filtering based on suspect channels classification. In *Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 419–426, Barcelona, Spain, 2012.

[117] Kristina Kunert, Magnus Jonsson, and Urban Bilstrup. Deterministic real-time medium access for cognitive industrial radio networks. In *Proceedings of the IEEE International Conference on Factory Communication Systems (WFCS)*, pages 91–94, Lemgo/Detmold, Germany, 2012.

[118] Yi Song and Jiang Xie. QoS-based Broadcast Protocol Under Blind Information in Cognitive Radio Ad Hoc Networks. In *Broadcast Design in Cognitive Radio Ad Hoc Networks*, pages 13–36. Springer, 2014.

[119] Pedro Manuel Rodríguez, Iñaki Val, Aitor Lizeaga, and Mikel Mendicute. Evaluation of cognitive radio for mission-critical and time-critical WSAN in industrial environments under interference. In *Proceedings of the IEEE International Conference on Factory Communication Systems (WFCS)*, Palma de Mallorca, Spain, 2015.

[120] Leonardo De Moura and Nikolaj Bjørner. Satisfiability Modulo Theories: Introduction and Applications. *Communications of the ACM*, 54(9):69–77, 2011.

[121] Luxi Zhao, Paul Pop, Zhong Zheng, and Qiao Li. Timing Analysis of AVB Traffic in TSN Networks using Network Calculus. In *Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 25–36, Porto, Portugal, 2018.

[122] Hyung-Taek Lim, Daniel Herrscher, Lars Völker, and Martin Johannes Waltl. IEEE 802.1AS time synchronization in a switched Ethernet based in-car network. In *Proceedings of the IEEE Vehicular Networking Conference (VNC)*, pages 147–154, Amsterdam, Netherlands, 2011.

[123] Annette Bohm, Magnus Jonsson, and Elisabeth Uhlemann. Performance Evaluation of a Platooning Application Using the IEEE 802.11p MAC on a Control Channel Vs. a Centralized Real-Time MAC on a Service Channel. In *Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 545–552, Lyon, France, 2013.

[124] András Varga. OMNeT++ Discrete Event Simulation System. In *Proceedings of the European Simulation and Modelling Conference (ESM)*, Prague, Czech Republic, 2001.

[125] Levente Mészáros, Andras Varga, and Michael Kirsche. INET Framework. In *Recent Advances in Network Simulation*, pages 55–106. Springer, 2019.

[126] Q. Zhao, L. Tong, A. Swami, and Y. Chen. Decentralized Cognitive MAC for Opportunistic Spectrum Access in Ad Hoc Networks: A POMDP Framework. *IEEE Journal on Selected Areas in Communications*, 25(3):589–599, 2007.

[127] Ivan Dominguez-Jaimes, Lukasz Wisniewski, Henning Trsek, and Jurgen Jasperneite. Link-Layer Retransmissions in IEEE 802.11g based Industrial Networks. In *Proceedings of the IEEE International Conference on Factory Communication Systems (WFCS)*, pages 83–86, Nancy, France, 2010.

[128] Magnus Jonsson and Kristina Kunert. Towards reliable wireless industrial communication with real-time guarantees. *IEEE Transactions on Industrial Informatics*, 5(4):429–442, 2009.

[129] Andreas Willig and Elisabeth Uhlemann. Deadline-aware scheduling of cooperative relayers in TDMA-based wireless industrial networks. *Wireless networks*, 20:73–88, 2014.

[130] Douglas Dimi Demarch and Leandro B. Becker. An Integrated Scheduling and Retransmission Proposal for Firm Real-Time Traffic in IEEE 802.11e. In *Proceedings of the Euromicro Conference on Real-Time Systems (ECRTS)*, pages 146–158, Pisa, Italy, 2007.

[131] Philip Parsch and Alejandro Masrur. A Reliability-Aware Medium Access Control for Unidirectional Time-Constrained WSNs. In *Proceedings of the International Conference on Real-Time Networks and Systems (RTNS)*, pages 297–306, Lille, France, 2015.

[132] Achim Berger, Albert Potsch, and Andreas Springer. TDMA proposals for wireless sensor networks for highly reliable and energy efficient data collection in an industrial application. In *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Kraków, Poland, 2012.

[133] Gianluca Cena, Stefano Scanzio, Lucia Seno, and Adriano Valenzano. A soft real-time scheduling framework for wireless industrial sensor actuator networks. In *Proceedings of the IEEE Symposium on Industrial Embedded Systems (SIES)*, Kraków, Poland, 2016.

[134] Gianluca Cena, Stefano Scanzio, Lucia Seno, and Adriano Valenzano. Optimal retransmission allocation for EDF-based networked real-time applications. In *Proceedings of the IEEE International Conference on Factory Communication Systems (WFCS)*, Sundsvall, Sweden, 2019.

[135] Jian Ma, Dong Yang, Hongchao Wang, and Mikael Gidlund. An Efficient Retransmission Scheme for Reliable End-to-End Wireless Communication Over WSANs. *IEEE Access*, 6:49838–49849, 2018.

[136] Cesare Alippi and Luigi Sportiello. Robust hybrid wired-wireless sensor networks. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pages 462–467, Mannheim, Germany, 2010.

[137] Samuele Zoppi, Amaury Van Bemten, H. Murat Gürsu, Mikhail Vilgelm, Jochen Guck, and Wolfgang Kellerer. Achieving Hybrid Wired/Wireless Industrial Networks With WDetServ: Reliability-Based Scheduling for Delay Guarantees. *IEEE Transactions on Industrial Informatics*, 14(5):2307–2319, 2018.

[138] Yu Nakayama, Kazuki Maruta, Takuya Tsutsumi, and Kaoru Sezaki. Wired and Wireless Network Cooperation for Wide-Area Quick Disaster Recovery. *IEEE Access*, 6:2410–2424, 2017.

[139] Giuseppe Bianchi. Performance Analysis of the IEEE 802.11 Distributed Coordination Function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, 2000.

[140] William Stallings. *Data and Computer Communications*. Pearson Education, Upper Saddle River, USA, 2007.

Appendices

Appendix A

Description of pseudocode procedures

Procedure	Description
channel_info ←**check_channel**(i)	Checks the channel i for interference during a certain interval and returns the result in *channel_info*.
is_broken ←**check_phy**($[v_i, v_j]$)	Indicates in *is_broken* whether the physical layer for link $[v_i, v_j]$ is broken or not.
avbl_channels ←**create_avbl_channels_list** (*channels_info*)	Creates a list of available channels *avbl_channels* from the information contained in *channels_info*.
m ← **dequeue_be**()	Retrieves the first message m from the FIFO queue that stores BE messages in the given node.
m ← **dequeue_critical**()	Retrieves the first message m from the FIFO queue that stores critical messages in the given node.
m ← **dequeue_critical_retx**()	Retrieves the first message m from the FIFO queue that stores critical retransmission messages the given node.
enqueue_critical_retx(m)	Enqueues the critical message m for retransmission in the FIFO queue.
is_corrupt ←**is_avbl_channels_corrupt** (*avbl_channels*)	Indicates in *is_corrupt* whether the information from available channels *avbl_channels* is corrupt and cannot be used.

Procedure	**Description**
is_empty ←**is_be_queue_empty**()	Indicates in *is_empty* whether the queue that stores BE messages in the node is empty.
is_critical ←**is_critical**(m)	Indicates in *is_critical* whether the message m is critical.
is_empty ←**is_critical_queue_empty**()	Indicates in *is_empty* whether the queue that stores critical messages in the node is empty.
is_empty ←**is_critical_retx_queue_empty**()	Indicates in *is_empty* whether the queue that stores critical retransmission messages in the node is empty.
is_be ←**is_current_slot_be**()	Indicates in *is_be* whether the current slot or the current phase is allocated for handling BE traffic.
is_critical ←**is_current_slot_critical**()	Indicates in *is_critical* whether the current slot is allocated for handling critical traffic.
is_critical_retx ←**is_current_slot_critical_retx**()	Indicates in *is_critical_retx* whether the current slot is allocated to handle retransmissions of critical traffic.
is_restricted ←**is_restricted_phase**()	Indicates in *is_restricted* whether the current point in time is within a restricted phase in which no transmissions can begin, but ongoing ones are allowed to finish.
mac_bh_1()	MAC procedure handling traffic according to `MAC_BH_1`. The pseudocode for the procedure is provided in Algorithm 8.2.
mac_bh_2_and_3b()	MAC procedure handling traffic according to `MAC_BH_2` or `MAC_BH_3b`. The pseudocode for the procedure is provided in Algorithm 8.3.
mac_bh_3a()	MAC procedure handling traffic according to `MAC_BH_3a`. The pseudocode for the procedure is provided in Algorithm 8.4.
mac_ch()	MAC procedure handling traffic according to `MAC_CH`. The pseudocode for the procedure is provided in Algorithm 8.1.

Procedure	Description
mac_cr_ap$_i$()	MAC procedure handling the cognitive radio protocol according to `MAC_CR` in an AP node i. The pseudocode for the procedure is provided in Algorithm 8.5.
mac_cr_es$_i$()	MAC procedure handling the cognitive radio protocol according to `MAC_CR` in an end-system node i. The pseudocode for the procedure is provided in Algorithm 8.5.
mac_sd_dyn_es$_i$()	MAC procedure handling the dynamic redundancy scheme according to `MAC_SD-_Dyn` on an end system node i. The pseudocode for the procedure is provided in Algorithm 9.4.
mac_sd_dyn_sw$_i$()	MAC procedure handling the dynamic redundancy scheme according to `MAC_SD-_Dyn` on a switch node i. The pseudocode for the procedure is provided in Algorithm 9.4.
mac_sd_sta_es$_i$()	MAC procedure handling the static redundancy scheme according to `MAC_SD-_Sta` on an end system node i. The pseudocode for the procedure is provided in Algorithm 9.3.
mac_sd_sta_sw$_i$()	MAC procedure handling the dynamic redundancy scheme according to `MAC_SD-_Dyn` on a switch node i. The pseudocode for the procedure is provided in Algorithm 9.3.
mac_td_1()	MAC procedure handling traffic according to `MAC_TD_Cons_1` or `MAC_TD_Sprd-_1`. The pseudocode for the procedure is provided in Algorithm 9.1.
mac_td_2()	MAC procedure handling traffic according to `MAC_TD_Cons_2` or `MAC_TD_Sprd-_2`. The pseudocode for the procedure is provided in Algorithm 9.2.
backoff_time ←**rand_inside_cw**()	Selects a random value in the *CW* range and returns it in *backoff_time*.
avbl_channels ←**recv_avbl_channels_from_ap**()	Blocking call to receive the list of available channels *avbl_channels* from the AP.

Procedure	**Description**
channels_info ←**recv_channel_info_from_ess**()	Blocking call to receive the *channels_info* from the end systems.
$dm_{k,z}$ ←**recv_from_app**()	Receives a data message $dm_{k,z}$ from the application layer.
m ←**recv_from_link**()	Receives a data message m from a link.
$\{dm_{k,z}, [v_x, v_i]\}$ ←**recv_from_link**()	Receives a data message $dm_{k,z}$ from the link $[v_x, v_i]$.
send_ack(m)	Sends an acknowledgment for the message m via a link.
send_asap(m)	Transmit a message m via the channel as soon as possible.
send_asap($dm_{k,z}$, $[v_i, v_j]$)	Sends a message $dm_{k,z}$ via the link $[v_i, v_j]$ as soon as possible.
send_avbl_channels_to_ess (*avbl_channels*)	Sends the list of available channels *avbl_channels* to the end systems via the channel.
remaining_backoff_time ←**send_be** (*aifs*, *backoff_time*)	Transmits over the channel the next BE message from the BE queue in the node. The channel is sensed for a duration of *aifs* and an additional *backoff_time*. A random *backoff_time* is selected if the provided value is 0. The remaining backoff time *remaining_backoff_time* is returned after subtracting the channel sensing time. The pseudocode for the procedure is provided in Algorithm 8.3.
send_channel_info_to_ap (*channel_info*)	Sends the retrieved *channel_info* to the AP via the channel.
remaining_backoff_time ←**send_csma_ca** (m, *aifs*, *backoff_time*)	Transmits a message m over the channel using the CSMA/CA procedure. The channel is sensed for a duration of *aifs* and an additional *backoff_time*. The remaining backoff time *remaining_backoff_time* is returned after subtracting the channel sensing time. The pseudocode for the procedure is provided in Algorithm 8.3.
send_lsm(*lsm*)	Sends a link-state message *lsm* via the link in the reverse path as soon as possible. The pseudocode for the procedure is provided in Algorithm 9.4.

Procedure	**Description**
send_scheduled$(dm_{k,z}, [v_i, v_j])$	Sends a message $dm_{k,z}$ over the link $[v_i, v_j]$ according to a schedule.
send_to_app$(dm_{k,z})$	Sends a data message $dm_{k,z}$ to the application layer.
is_detected ←**sense_channel**(*duration*)	Senses the communication channel for the given *duration* and indicates in *is_detected* whether a transmission was detected.
switch_tx_channel(*tx_channel*)	Switches the transmission channel to *tx_channel*.
remaining_backoff_time ←**update_backoff**(*backoff_time*)	Update the *backoff_time* to reflect the remaining backoff time *remaining_backoff_time* after sensing the channel.
is_recv ←**wait_for_ack**(m)	Blocking call to wait for the acknowledgment of m for a certain time. Returns in *is_recv* whether the ACK is received.
wait_for_assigned_slot()	Blocking call to wait for the next time slot assigned to the given node.
wait_for_cc()	Blocking call to wait for the next cognitive radio control (CC) phase in cognitive radio.
wait_for_msg()	Blocking call to wait for incoming messages, either from the application layer or the link.
wait_for_ss()	Blocking call to wait for the next spectrum sensing (SS) phase in cognitive radio.

Appendix B

Acronyms

6P	6top protocol
ABS	Anti-lock braking system
AC	Access category
ACK	Acknowledgment or acknowledgment frame
ACL	Asynchronous connectionless
AFDX	Avionics Full-Duplex Switched Ethernet
AIFS	Arbitration IFS
AP	Access point STA
ARQ	Automatic repeat request
ASIL	Automotive Safety and Integrity Levels
AWGN	Additive white Gaussian noise
BE	Best effort
BMCA	Best master clock algorithm
BSS	Basic service set
CA	Collision avoidance
CAN	Controller area network
CAN-FD	CAN Flexible Data Rate
CAP	Controlled-access phase
CBS	Credit-based shaper
CC	Cognitive radio control
CCC	Common control channel
CD	Collision detection
CF-Poll	Contention-free poll
CFP	Contention-free period

CN	Controlled node
COTS	Commercial off-the-shelf
CP	Contention period
CR	Cognitive radio
CRC	Cyclic redundancy check
CSMA	Carrier sense multiple access
CTS	Clear to send
CW	Contention window
DAL	Design Assurance Level
DCF	Distributed coordination function
DCU	DCF unit
DIFS	DCF IFS
E2E	End-to-end
ECU	Electronic control unit
EDCA	Enhanced distributed channel access
EDF	Earliest-deadline first
ESS	Extended service set
EtherCAT	Ethernet for Control Automation Technology
FFD	Full function device
FH	Frequency hopping
FHSS	Frequency-hopping spread spectrum
FIFO	First in, first out
FOL	First-order logic
FP	Fixed priority
FPGA	Field-programmable gate array
FTT-E	Flexible Time-Triggered Ethernet
GCL	Gate control list
gPTP	Generalized precision time protocol
GTS	Guaranteed time slot
HART	Highway Addressable Remote Transducer
HCCA	Hybrid coordination function coordinated channel access
HeNReS	Heterogeneous Hybrid Networks Reliability Simulation
HSR	High-availability Seamless Redundancy
IBSS	Independent basic service set
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force

IFS	Interframe space
ILP	Integer linear programming
IoT	Internet of things
IP	Internet Protocol
IRT	Isochronous real-time
IS-IS	Intermediate System - Intermediate System
ISM	Industrial, scientific and medical
ISO	International Organization for Standardization
IT	Information technology
LAN	Local area network
LLC	Logical link control
LSDB	Link State Database
LSP	Link State PDU
MAC	Medium/media access control
MIMO	Multiple-input and multiple-output
MIT	Minimum inter-arrival time
MLME	Media Access Control Sublayer Management Entity
MN	Managed node
MRP	Media Redundancy Protocol
MSF	Minimal scheduling function
NACK	Negative acknowledgment
NAV	Network allocation vector
NED	Network Description language
NIC	Network interface card
NTP	Network Time Protocol
OS	Operating system
OSI	Open Systems Interconnection
OT	Operational technology
P2P	Peer-to-peer
PAN	Personal area network
PCF	Point coordination function
PCP	Priority code point
PCU	Protocol control unit
PDU	Protocol data unit
PHY	Physical layer
PIFS	PCF IFS
PLC	Programmable logic controller

PRP	Parallel Redundancy Protocol
PTP	Precision time protocol
QCU	Queue control unit
RBD	Reliability block diagrams
RC	Rate constrained
RDR	Request Data with Reply
RFD	Reduced function devices
RQ	Research question
RSTP	Rapid Spanning Tree Protocol
RT	Real-time
RTS	Request to send
SCO	Synchronous connection-oriented
SDA	Send Data with Acknowledge
SDN	Send Data with No acknowledge
SDN	Software-defined networking
SDR	Software-defined radio
SF	Scheduling Function
SIFS	Short IFS
SMT	Satisfiability modulo theories
SNP	Sequence Number PDU
SNR	Signal-to-noise ratio
SPB	Shortest Path Bridging
SRD	Send and Request Data
SRP	Stream Reservation Protocol
SS	Spectrum sensing
ST	Scheduled traffic
STA	Station (includes APs)
STP	Spanning Tree Protocol
TCP	Transmission Control Protocol
TDMA	Time-division multiple access
TSCH	Time-slotted channel hopping
TSF	Timing synchronization function
TSN	Time-Sensitive Networking
TT	Time-triggered
TT-CAN	Time-Triggered CAN
TTE	Time-Triggered Ethernet
TXOP	Transmission opportunity

UDP	User datagram protocol
VL	Virtual link
VLAN	Virtual LAN
WAN	Wide area network
WCET	Worst-case execution time
WIA-FA	Wireless Networks for Industrial Automation - Factory Automation
WISA	Wireless Interface for Sensors and Actuators
WSAN	Wireless sensor and actuator networks
WSN	Wireless sensor networks

Appendix C

Notations

(v_i, v_j)	Physical link. Undirected edge between v_i and v_j in the physical network topology.
$[v_i, v_j]$	Dataflow link. Directed edge between v_i and v_j in the physical network topology.
λ_{wd}	Rate at which wired dataflow links fail.
τ	Channel access probability of a single wireless system during T_{ST}
τ_i	Processing task i
Φ_i	Offset of τ_i or m_i
C_i	Execution time of τ_i or transmission time of m_i
CW	Contention window interval in CSMA/CA
CW_{min}	Lower boundary for the CW interval
CW_{max}	Upper boundary for the CW interval
D_i	Deadline for the execution of τ_i or transmission of m_i
$dm.paths$	Set of all wired paths for all dm_i
dm_i	Data message i. Note that m_i is also used for the same purpose depending on the described mechanism.
$dm_i.size$	Size of dm_i
$dm_i.path.active$	Indicates the state of $dm_i.path$: *True* means enabled and *false* means disabled.
$dm_i.path.path$	Ordered set of dataflow links conforming the path of dm_i : $([v_a, v_b], [v_b, v_c], ..., [v_d, v_e])$.
$dm_i.paths$	Set of wired paths for dm_i
$dm_{i,j}$	Instance j of dm_i
E	Set of (v_i, v_j) in the physical network topology
f_i	Arrival frequency of τ_i or m_i

$G(V, E)$	Physical network topology, conformed by V and E
$J_i^{handling}$	Handling jitter of τ_i or m_i
$J_i^{release}$	Release jitter of τ_i or m_i
$J_i^{response}$	Response time jitter of τ_i or m_i
L	Set of $[v_i, v_j]$ in the physical network topology
L_{wd}	Subset of the wired dataflow links included in L
L_{wl}	Subset of the wireless dataflow links included in L
lsm	Link-state message
$lsm.broken\_link$	Field from the lsm indicating whether a link is broken.
$lsm.size$	Size of the lsm
M	Set of m_i
m	Data message
m_i	Data message i. Note dm_i is also used for the same purpose, depending on the described mechanism.
$m_i.n$	Number of times, i.e. copies, m_i is transmitted.
$m_i.size$	Size of m_i
$m_i^{[v_j, v_k]}.n$	Number of times, i.e. copies, m_i is transmitted via $[v_j, v_k]$.
$m_{i,j}$	Instance j of m_i
$m_{i,j}^{[v_k, v_l]}$	Transmission of $m_{i,j}$ via $[v_k, v_l]$
$O_i^{[v_k, v_l]}$	Instant, referring to the start of the schedule, at which $m_{i,1}$ is scheduled for transmission via $[v_k, v_l]$.
$O_{i,j}^{[v_k, v_l]}$	Instant, referring to the start of the schedule, when $m_{i,j}$ is scheduled for transmission via $[v_k, v_l]$.
R	Bitrate
$R_{dyn}^i(t)$	Reliability of the data message dm_i at instant t with the dynamic redundancy mechanism (`MAC_SD_Dyn`)
$R_{dyn\_wl}^i(t)$	Reliability of the data message dm_i at instant t transmitted via the wireless segment with the dynamic redundancy mechanism (`MAC_SD_Dyn`)
R_i	Response time of τ_i or m_i
$R_i(t)$	Reliability of the data message dm_i at instant t
$R_{none}^i(t)$	Reliability of the data message dm_i at instant t without redundancy mechanism (`MAC_SD_None`)
$R_{not\_col}^{i,j}(t)$	Probability that the data message dm_i does not collide with the data message dm_j at instant t on the wireless channel.
$R_{not\_col\_wl}(t)$	Probability that a wireless transmission under DCF does not collide at instant t.

$R_S(t)$	Reliability of the whole wired and wireless communication system at instant t with fault tolerance using space diversity mechanisms. The reliability value could represent the cases without redundancy (`MAC_SD_None`), with static redundancy (`MAC_SD_Sta`) or with dynamic redundancy (`MAC_SD_Dyn`).
$R^i_{sta}(t)$	Reliability of the data message dm_i at instant t with the static redundancy mechanism (`MAC_SD_Sta`)
$R^i_{sta\_wl}(t)$	Reliability of the data message dm_i at instant t transmitted via the wireless segment with the static redundancy mechanism (`MAC_SD_Sta`)
$R^i_{wd}(t)$	Reliability of the data message dm_i at instant t when transmitted via the wired segment.
S	Number of reserved time slots for critical traffic
T	Set of T_i
T_{AIFS}	Duration of the AIFS of IEEE 802.11
T^{CR}	Period of the cognitive radio phase
T_{CW}	Selected CW duration
T_{DIFS}	Duration of the DIFS of IEEE 802.11
T_{IFS}	Duration of the IFS of IEEE 802.11
T_{ITI}	Inter transmission interval. The time difference between transmissions and retransmissions of the same data message instance $m_{i,j}$. The value is the same for all messages m.
T_{SIFS}	Duration of the SIFS of IEEE 802.11
T_{ST}	Duration of the slot of IEEE 802.11
T_{ack}	Duration required for the transmission of an ACK message
T_{burst}	Duration of an interference burst
T_{data}	Duration required for the transmission of a data message m
T_{delay}	Channel access delay
T_i	Period of τ_i or m_i
T_i^{min}	Minimum inter-arrival time of τ_i or m_i
T_{slot}	Duration of a time slot
T_{slot}^{CR}	Duration of a cognitive radio slot
V	Set of v in the physical network topology
V_{wl}	Subset of V that are wireless nodes
v_i	Node i in the physical network topology
v_o	Refers to the wireless broadcast domain. Every link with v_o as destination reaches any $v \in V$ that has a wireless interface.
w_i	Weight that describes the influence of $R_i(t)$ on $R_S(t)$.
X_i	Criticality of τ_i or m_i

ISBN 978-91-7485-673-6
ISSN 1651-4238

Address:
P.O. Box 883, SE-721 23 Västerås. Sweden
P.O. Box 325, SE-631 05 Eskilstuna. Sweden
E-mail: info@mdu.se **Web:** www.mdu.se

Mälardalen
University