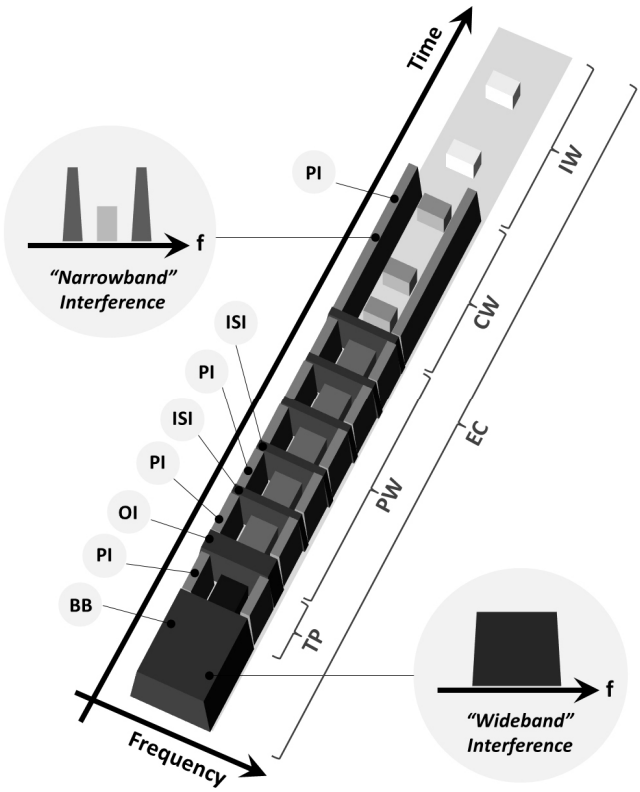
Universidade de Aveiro Departamento de Eletrónica, Telecomunicações e Informática
2014

Paulo Jorge de Campos Bartolomeu

Comunicações Sem-Fios de Tempo-Real para Ambientes Abertos

Dependable Wireless Real-Time Communications for Open Environments



Universidade de Aveiro
2014

Departamento de Eletrónica, Telecomunicações e Informática

Paulo Jorge de Campos Bartolomeu

Comunicações Sem-Fios de Tempo-Real para Ambientes Abertos

Dependable Wireless Real-Time Communications for Open Environments

Tese apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Doutor em Engenharia Informática, realizada sob a orientação científica do Doutor José Alberto Gouveia Fonseca, Professor Associado do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro e co-orientação do Doutor Francisco Manuel Madureira e Castro Vasques de Carvalho, Professor Associado da Faculdade de Engenharia da Universidade do Porto.

Dissertation submitted to the University of Aveiro in fulfillment of the requirements for the degree of *Doutor em Engenharia Informática*, under the supervision of José Alberto Gouveia Fonseca, *Professor Associado* at the Departamento de Eletrónica, Telecomunicações e Informática of the University of Aveiro and co-supervision of Francisco Manuel Madureira e Castro Vasques de Carvalho, *Professor Associado* at the *Faculdade de Engenharia* of the University of Oporto.

Apoio financeiro da FCT e do FSE no âmbito do III Quadro Comunitário de Apoio

Apoio financeiro COMPETE - Programa Operacional Factores de Competitividade

Apoio financeiro da Micro I/O – Serviços de Electrónica, Lda.

Apoio financeiro Mais Centro – Programa Operacional do Centro

Dedico este trabalho às mulheres da minha vida
(Laurinda, Ângela, Patrícia e Beatriz)

o júri

presidente

Professor Doutor Nelson Fernando Pacheco da Rocha
Professor Catedrático da Universidade de Aveiro

Professor Doutor Dietmar Dietrich
Professor Catedrático da Universidade de Viena, Áustria

Professora Doutora Alessandra Flammini
Professora Associada da Universitá degli Studi di Brescia, Itália

Professor Doutor José Manuel de Sousa de Matos Rufino
Professor Auxiliar da Universidade de Lisboa

Professor Doutor Pedro Alexandre Guimarães Lobo Ferreira do Souto
Professor Auxiliar da Universidade do Porto

Professor Doutor Joaquim José de Castro Ferreira
Professor Adjunto da Universidade de Aveiro

Professor Doutor Francisco Manuel Madureira Vasques de Carvalho
Professor Associado da Universidade do Porto (Co-Orientador)

Professor Doutor José Alberto Gouveia Fonseca
Professor Associado da Universidade de Aveiro (Orientador)

agradecimentos

O trabalho realizado no âmbito desta tese contou com a colaboração, direta e indireta, de diversas pessoas. Quero aqui e a todas elas expressar o meu sincero agradecimento. Contudo e devido ao seu especial envolvimento, gostaria de particularizar os seguintes agradecimentos.

Esta tese não teria sido possível sem o apoio inquestionável do professor José Alberto Fonseca, orientador deste trabalho. Ele foi o responsável pela minha incursão no mundo da investigação científica. A ele devo não só a motivação para realizar um doutoramento em empresa e a supervisão científica desta tese, mas também o seu apoio profissional e pessoal ao longo da última década. Considero-me um previlegiado por ter tido a oportunidade de colaborar e de aprender com ele.

Agradeço, também, ao professor Francisco Vasques, co-orientador da tese, o impulso de abraçar este desafio. O seu estímulo, materializado na candidatura a uma bolsa de doutoramento da FCT, foi o ponto de viragem na decisão de focar o meu trabalho nas redes de comunicação sem fios para ambientes abertos. Estou, ainda, grato pelo seu acompanhamento ao longo de todo o percurso e pela colaboração em vários projetos de interesse comum.

I want also to express my gratitude to the Wireless Sensor Networks LABoratory Group of the Electronics for Automation Department of the University of Brescia. Their help in validating the implementation of a Programmable Interference Synthesizer at an early stage of this work was crucial for the developments that followed. I want to especially thank professors Alessandra Flammini and Paolo Ferrari for their openness in establishing this collaboration. To professor Paolo Ferrari and Dott. Ric. Chiara De Dominicis I owe the technical characterization and core implementation of the SDR based PNS.

Devo um agradecimento especial à Micro I/O – Serviços de Electrónica, Lda., empresa co-financiadora desta tese, pelas condições que me proporcionou e que foram imprescindíveis para a realização deste trabalho. Neste âmbito, quero referir o apoio financeiro dos projetos de I&D em que participei, nomeadamente aquele que foi obtido pelo projeto CIRaF – "Tecnologias para Comunicação e Identificação por Rádio Frequência", através do Programa Operacional do Centro; e pelos projetos "Living Usability Lab" e ACTOR – "Apostar na Certificação das Empresas Tice Organizadas em Rede", através do Programa Operacional do Centro e do Programa Operacional Factores de Competitividade. Agradeço também o apoio financeiro da Fundação para a Ciência e a Tecnologia obtido através da BDE e do projeto "DHT-Mesh: Serviços baseados em DHT para melhorar a escalabilidade de Redes em Malha com Alta disponibilidade".

agradecimentos

A equipa da Micro I/O merece um agradecimento particular pela amizade, pelo incentivo e pela confiança neste trabalho. Em particular, agradeço ao Engenheiro Fernando Almeida, diretor da empresa, todo o apoio durante a condução deste trabalho. As condições que me facultou foram determinantes para ter chegado até aqui. De forma direta ou indireta todos me ajudaram. Contudo, devo um agradecimento especial à Andreia Abreu pelo apoio no desenvolvimento do PNS; à Maria Inês pela ajuda na implementação da aplicação WITAS em Java; e ao Vasco Baptista pelo contributo no desenvolvimento do BeeMon e do WFTT. Não posso deixar, também, de registar a amizade e o incentivo persistente de vários colaboradores da empresa, especialmente do André Peixoto, da Maria João, do Mário Couto, do Daniel Silva e do Ricardo Almeida.

O apoio e o incentivo dos meus amigos do DETI foram também determinantes para chegar a este ponto. Ao longo da última década tive o previlégio de conviver com inúmeros professores e investigadores deste departamento com quem partilhei muitas horas de convívio e amizade. Estou grato pelas suas críticas construtivas e pelo encorajamento. Em particular, quero agradecer a amizade e o apoio do Alexandre Mota, Arnaldo Oliveira, Frederico Santos, Joaquim Ferreira, Luís Almeida, Ricardo Marau, Paulo Pedreiras, Margarida Urbano, Rui Santos e Valter Silva. Ao Joaquim Ferreira agradeço, também, a revisão técnica desta tese, que muito contribuiu para o seu melhoramento.

Tenho também de agradecer o apoio da minha família. Os meus avós desde cedo me incentivaram a chegar mais longe, a melhorar e a superar-me. O meu avô Arsénio nunca deixou de me encorajar a concluir este desafio. Os meus pais sempre acreditaram no meu potencial. Ao meu pai António devo a oportunidade de ter realizado um curso superior universitário, pois foi ele quem insistiu em dar-me essa oportunidade. No início do meu percurso académico foram muitos os sacrifícios que fizeram para que eu tivesse condições para prosseguir os meus estudos. Todos me apoiaram sem nunca questionar a minha ausência e as minhas opções. Os meus sucessos são os sucessos deles também.

Finalmente, estou grato pelo apoio da Patrícia e do meu irmão Nuno. Estarei para sempre em dívida com ela por ser a melhor parte de mim. Foi a amiga que precisava nos momentos difíceis, a professora que me ajudou na revisão da tese e a esposa que, para me dar tempo para concluir este desafio, assumiu muitas vezes sozinha o cuidado da Beatriz. Sem o seu apoio teria sido impossível concluir este trabalho.
Ao meu melhor amigo e irmão agradeço o apoio incondicional e constante ao longo destes anos. Sempre fomos uma equipa e continuaremos a sê-lo sempre.

palavras-chave

Comunicações sem-fios, sistemas distribuídos, sistemas embutidos, acesso ao meio, captura de banda, tempo-real, ambientes abertos, fiabilidade, *black-burst*.

resumo

As tecnologias de comunicação sem fios tornaram-se amplamente adoptadas, surgindo em aplicações heterógeneas que vão desde a localização de vítimas, pessoal médico e equipamentos em cenários de desastre à monitorização da condição física de máquinas em ambientes industrials. Muito frequentemente, as aplicações exigem uma resposta limitada no tempo que, geralmente, em sistemas distribuídos, é substancialmente dependente do desempenho da tecnologia de comunicação utilizada. Estes sistemas tendem a possuir requisitos de tempo-real uma vez que a comunicação de dados tem de ser conduzida dentro de limites temporais pré-definidos que, quando não cumpridos, podem comprometer o correcto funcionamento do sistema e resultar em perdas económicas ou colocar em risco vidas humanas.

A potencial adopção de tecnologias sem-fios para um crescente número de cenários traduz-se num aumento da complexidade e heterogeneidade dos requisitos operacionais relativamente às tecnologias cabladas. A acompanhar esta tendência verifica-se uma crescente procura de sistemas distribuídos, caracterizados quer por uma boa relação custo-eficácia, quer pela simplicidade de instalação, manutenção e adaptação. Ao mesmo tempo, estes sistemas tendem a requerer flexibilidade operacional, que apenas pode ser assegurada se a tecnlogia de comunicação empregue supportar transmissões de dados dispoletadas quer por eventos (*event-triggered*), quer por tempo (*time-triggered*) e se, ao mesmo tempo, em funcionamento, permitir a alteração dos parâmetros de comunicação correspondentes.

Frequentemente, as aplicações com comunicações sem fios caracterizam-se por exigências de instalação que apenas podem ser endereçadas usando alimentação através de baterias e/ou mecanismos de recolha de energia do ambiente envolvente. Estas aplicações têm tipicamente requisitos exigentes de autonomia e de tamanho, impedindo o recurso a baterias de grande dimensão. Dado que o suporte de comunicações pode representar uma parte significativa dos requisitos de energia da estação, o uso de tecnologias de comunicação de elevado consumo não é adequado. Desta forma, nestas aplicações, as tecnologias de comunicação de curto-alcance tornaram-se amplamente adoptadas uma vez que, apesar de se caracterizarem por taxas de transmissão inferiores, consomem apenas uma fracção da energia das tecnologias de maior alcance.

resumo

Em geral, os requisitos de pontualidade da comunicação de dados podem ser cumpridos através da garantia da disponibilidade do meio no instante em que qualquer estação inicie uma transmissão. Em ambientes controlados esta disponibilidade pode ser garantida, na medida em que existe um controlo de quais as estações que foram instaladas na área e qual a sua função. Contrariamente, em ambientes abertos, tal controlo é difícil de garantir uma vez que não existe conhecimento *a priori* de que estações ou tecnologias podem competir pelo meio, tornando o suporte de comunicações de tempo-real um desafio difícil de implementar em cenários com estações de comunicação não controladas.

As comunicações de baixo consumo têm sido o foco de um esforço de investigação bastante amplo, por exemplo, no domínio das redes de sensores sem fios. Embora possam permitir uma maior autonomia a estações baseadas em baterias, estas tecnologias são reconhecidas como sendo negativamente influenciadas por tecnologias semelhantes competindo pelo mesmo meio e, em particular, por tecnologias que utilizem níveis de potência de transmissão mais elevados em bandas de frequências comuns. De forma cada vez mais acentuada, a banda industrial, científica e médica (ISM) dos 2.4 GHz tem-se tornado mais saturada com tecnologias que competem entre si pelo acesso ao meio tais como, por exemplo, Bluetooth e ZigBee, dois padrões de comunicação que são a base de vários protocolos de tempo-real. Apesar destas tecnologias aplicarem mecanismos para melhorar a sua coexistência, são vulneráveis a transmissões de estações não controladas que usem as mesmas tecnologias ou que usem tecnologias com níveis de potência de transmissão mais elevados, impedindo, desta forma, o suporte de comunicações de tempo-real fiáveis em ambientes abertos.

O protocolo de comunicação sem fios flexível disparado por tempo (WFTT) é baseado numa arquitectura mestre/múltiplo escravo alavancado na flexibilidade e pontualidade promovidas pelo paradigma FTT e na captura e manutenção determinística do meio suportadas pela técnica de *bandjacking* (captura de banda). Esta tese apresenta o protocolo WFTT e argumenta que este permite suportar serviços de comunicação de tempo-real com requisitos elevados de fiabilidade em ambientes abertos onde várias tecnologias de comunicação baseadas em contenção disputam o acesso ao meio.

resumo

Adicionalmente, esta tese reivindica que é possível suportar comunicações sem-fios simultaneamente flexíveis e pontuais em ambientes abertos. O protocolo WFTT foi inspirado no paradigma FTT, do qual importa os serviços de alto nível como, por exemplo, o controlo de admissão. Após a observação da eficácia da técnica de *bandjacking* em assegurar o acesso ao meio e a correspondente manutenção, foi reconhecida a possibilidade de utilização deste mecanismo para o desenvolvimento de um protocolo de acesso ao meio, capaz de oferecer as funcionalidades do paradigma FTT em meios de comunicação sem-fios. O desempenho do protocolo WFTT é reportado nesta tese com uma descrição dos dispositivos implementados, da bancada de ensaios desenvolvida e dos resultados obtidos.

keywords

Wireless communications, distributed systems, embedded systems, medium access, bandjacking, real-time, open environments, dependability, black-burst.

abstract

Wireless communication technologies have become widely adopted, appearing in heterogeneous applications ranging from tracking victims, responders and equipments in disaster scenarios to machine health monitoring in networked manufacturing systems. Very often, applications demand a strictly bounded timing response, which, in distributed systems, is generally highly dependent on the performance of the underlying communication technology. These systems are said to have real-time timeliness requirements since data communication must be conducted within predefined temporal bounds, whose unfulfillment may compromise the correct behavior of the system and cause economic losses or endanger human lives.

The potential adoption of wireless technologies for an increasingly broad range of application scenarios has made the operational requirements more complex and heterogeneous than before for wired technologies. On par with this trend, there is an increasing demand for the provision of cost-effective distributed systems with improved deployment, maintenance and adaptation features. These systems tend to require operational flexibility, which can only be ensured if the underlying communication technology provides both time and event triggered data transmission services while supporting on-line, on-the-fly parameter modification.

Generally, wireless enabled applications have deployment requirements that can only be addressed through the use of batteries and/or energy harvesting mechanisms for power supply. These applications usually have stringent autonomy requirements and demand a small form factor, which hinders the use of large batteries. As the communication support may represent a significant part of the energy requirements of a station, the use of power-hungry technologies is not adequate. Hence, in such applications, low-range technologies have been widely adopted. In fact, although low range technologies provide smaller data rates, they spend just a fraction of the energy of their higher-power counterparts.

The timeliness requirements of data communications, in general, can be met by ensuring the availability of the medium for any station initiating a transmission. In controlled (close) environments this can be guaranteed, as there is a strict regulation of which stations are installed in the area and for which purpose. Nevertheless, in open environments, this is hard to control because no *a priori*

abstract

knowledge is available of which stations and technologies may contend for the medium at any given instant. Hence, the support of wireless real-time communications in unmanaged scenarios is a highly challenging task.

Wireless low-power technologies have been the focus of a large research effort, for example, in the Wireless Sensor Network domain. Although bringing extended autonomy to battery powered stations, such technologies are known to be negatively influenced by similar technologies contending for the medium and, especially, by technologies using higher power transmissions over the same frequency bands. A frequency band that is becoming increasingly crowded with competing technologies is the 2.4 GHz Industrial, Scientific and Medical band, encompassing, for example, Bluetooth and ZigBee, two low-power communication standards which are the base of several real-time protocols. Although these technologies employ mechanisms to improve their coexistence, they are still vulnerable to transmissions from uncoordinated stations with similar technologies or to higher power technologies such as Wi-Fi, which hinders the support of wireless dependable real-time communications in open environments.

The Wireless Flexible Time-Triggered Protocol (WFTT) is a master/multi-slave protocol that builds on the flexibility and timeliness provided by the FTT paradigm and on the deterministic medium capture and maintenance provided by the *bandjacking* technique. This dissertation presents the WFTT protocol and argues that it allows supporting wireless real-time communication services with high dependability requirements in open environments where multiple contention-based technologies may dispute the medium access. Besides, it claims that it is feasible to provide flexible and timely wireless communications at the same time in open environments. The WFTT protocol was inspired on the FTT paradigm, from which higher layer services such as, for example, admission control has been ported. After realizing that *bandjacking* was an effective technique to ensure the medium access and maintenance in open environments crowded with contention-based communication technologies, it was recognized that the mechanism could be used to devise a wireless medium access protocol that could bring the features offered by the FTT paradigm to the wireless domain. The performance of the WFTT protocol is reported in this dissertation with a description of the implemented devices, the test-bed and a discussion of the obtained results.

Contents

List of Figures

List of Tables

LIST OF TABLES

"I believe there is no philosophical high-road in science, with epistemological signposts. No, we are in a jungle and find our way by trial and error, building our road behind us as we proceed."

Max Born (1882 - 1970)

1 Introduction

Over the last few years, low-power wireless communication technologies have experienced an increasing popularity in a wide range of applications. Although they provide a valuable set of benefits such as flexibility and cost-effectiveness, they also present some limitations when compared to their wired counterparts (interference, security, etc.). Nevertheless, motivated by their ability to perform short-range transmissions with a reduced amount of power, several applications have appeared. Notably, this set of applications is highly heterogeneous both in terms of the target domains (military, medical, civil, etc.) and of their requirements (best-effort, soft real-time, hard real-time, etc.). In the following section, an overview of the most relevant applications is conducted with focus on identifying and quantifying their communication requirements. Following, the core problems addressed by this dissertation and the contributions for a solution are defined. Finally, an outline of this dissertation is presented.

1.1 Selected Applications

Wireless communications have found an increasing number of applications. In particular, short-range communications have managed to penetrate into application domains such as the consumer goods industry, given their flexibility, scalability, mobility support and low-power operation [1]. Furthermore, besides reducing installation costs, Wireless Personal Area Networks (WPANs) have been used to mitigate problems related to the deployment of sensors in dangerous places or having a highly difficult access.

Although applications may share common features such as the networking topology or the use of a given standard technology, the demands concerning the wireless network can be highly heterogeneous. On one end, some applications aim at supporting real-time

communication services in a predictable and reliable fashion so that guarantees can be provided regarding the delivery of packets within bounded times. On the other, applications can be supported on best-effort communication services with loose requirements regarding timeliness and reliability. Three groups of applications were selected for analysis: localization, monitoring and synchronization. Together with their adoption of short-range wireless communications, this selection was motivated by the application's potential operation in spaces shared with different wireless technologies. The localization of people in indoor environments is often supported on wireless technologies allowing to track patients in hospitals, for example. Wireless enabled environmental and health monitoring can be found in domotics and health-care applications. In the first, temperature and humidity monitoring may allow a better control of the environment conditions and the provision of an improved comfort level. In the second, using physiological sensors, health monitoring may be used to detect the onset of a disease, allowing to trigger a more timely response by the caregiver. Finally, synchronization dependent applications enable strict timing operations such as distributed sensor sampling, for example. This is important in scenarios where data is to be acquired in different locations at the same time such as, for example, in the field of seismic monitoring. In the following subsections an overview of these key applications is provided, as well as of the associated requirements.

1.1.1 Localization Applications

Localization refers to the position estimation of an asset in a given area and it is a promising application for wireless communication technologies. Although existing commercial solutions are mainly based on Wireless Local Area Network (WLAN) technologies such as IEEE 802.11 (Ekahau RTLS [2], Exavera eShepherd [3], Aeroscout Real-time Visibility [4], Ubisense Precise Real-time Location [5]), short-range communications are becoming increasingly popular for indoor tracking due to their reduced power consumption, cost-effectiveness, reliability and coverage.

Short-range based localization based on *Receiver Signal Strength Indication* (RSSI) can be found in both public spaces (e.g., hospitals and museums) or private settings (e.g., factories or houses). The provision of personalized multimedia contents to the visitors in a museum [6] is a localization application where the users of interest are given a customized tour and tracked within the exhibitions. The use of a localization mechanism enables a more personalized user experience, e.g., via a follow-up email with relevant information regarding the objects that arose more interest in the exhibition. Furthermore, localization

allows improving the management of the exhibitions. This can be achieved by tracking which exhibitions and objects are popular and by replacing those who are not, for example. Similarly, at home, context-aware applications and services can be provided according to the localization of its inhabitants [7]. For example, if a localization system provides the identity of the user seeing a movie, such information combined with the knowledge that the user suffers from myopia can trigger an adaptation of the subtitle's font size and, consequently, improve the user's multimedia experience. Another example where RSSI based ranging can be used is in underground wireless networks to detect seismic activity [8]. In this application, wireless sensors are displaced on the subsurface of the area of interest and, whenever a seismic event changes the relative distance between sensor nodes, the signal strength "signature" changes, thus allowing to detect the event. This method can be used to provide an early detection of the onset of geological events such as earthquakes.

Emergency response is another emerging application that builds on short-range wireless communications to track victims, responders and equipment in disaster scenarios [9]. Two examples of this application are firefighting [10, 11] and response to chemical explosions [12]. In these scenarios, the localization of the firefighters is critical not only for the logistic coordination of large operations in unknown environments, but also to improve the tactical response in the advent of unforeseen occurrences. Moreover, complemented with real-time environment and health data, localization can play a crucial role in improving the safety of the responders and victims in demanding and chaotic environments such as the ground zero in New York during the terrorist attacks of the 9/11, for example.

Localization can also improve resource management and logistics by allowing a real-time position monitoring of assets over large areas such as warehouses and production plants [13]. In this application, all objects (boxes, parts, office equipment, etc.) are tagged with sensors during their check-in. Besides monitoring storage conditions (e.g. temperature) to improve the control of the environment conditions, for instance, using a HVAC system, tags also report their position. This allows tracking mobile equipment and prevent its loss or theft. Besides, the knowledge of the equipments' localization substantially reduces the need for redundancy in scenarios where such equipments need to be quickly found.

The detection (or response) of (to) health and safety violations in working facilities can also be improved by localization services. In this scenario, the workers' localization is monitored in harsh environments (e.g. in a mine) or conducting stressful tasks for limited periods of time [14]. When the workers' exposure goes beyond the threshold defined by the health and safety regulations, an alert is generated, allowing to notify the worker

and his/her supervisor. Another example where localization plays an important role is when a task, for security reasons, requires a specific number of workers in a geographical area. In this case, to avoid violating health and safety regulations, the employed tools can be programmed to operate if - and only if - the minimum number of workers wearing localization tags are in that area.

Robot navigation is yet another application where localization plays an important role allowing robots to perform the irrigation, lawn mowing or fertilizing of large areas [15], for example. In this case, sensors are commonly deployed in the garden to enable applications such as plant monitoring, weather forecast and/or surveillance of the area. Hence, the robot may take advantage of the wireless infrastructure-like network of sensors, installed in predefined spots, to estimate its own localization based on RSSI, thus improving its navigation and performing a better lawn mowing or fertilization of the garden.

1.1.2 Monitoring Applications

Monitoring refers to the observation of a given variable set and it has been one of the most studied areas of application for short-range wireless communications. Applications span the monitoring of environment variables (e.g., temperature), vital signals (e.g., cardiac rhythm) and machine condition (e.g., vibration), among others. Wireless environmental monitoring is found in several scenarios such as HVAC control, sewage treatment, water quality monitoring and quality assurance in food transportation, for example. In the first case, sensors are placed in the stored items of interest (or nearby) to provide information, which helps improving the control of the heating, ventilation, and air conditioning of the warehouse [13]. Water quality monitoring and treatment are two critical aspects of any densely populated area because they have a huge impact on the health of the population. The water quality can be monitored in real-time using a wireless sensor network (WSN), which continuously collects information about the waters' PH value, pollutant levels, temperature and turbidity [16]. Sewage treatment can be enabled by wireless monitoring technologies, which allow building automated plants that are more flexible and safer to maintain [17]. Regarding the transport of food, during the strawberries' journey from farm to the store, parameters such as the transport's environment temperature and humidity can be closely monitored, providing a higher degree of confidence to the store and allowing the transport company to act preventively when some unsuspected condition is detected. Hence, this monitoring process allows tracking and preventing problems in the supply chain, which promote the freshness and quality of the delivered products [18].

The wireless monitoring of industrial processes is another application where a given set of variables are continuously observed and can be used as inputs for controlling a given process [19]. One of these applications is found in chemical plants, where several flows of data "must be gathered by sensors distributed all around the plant" and communicated to the plant controller, which ensures its operation within specific safety bounds [20].

The monitoring of machine health condition can be performed in the context of a networked manufacturing system, for instance. By continuously observing the key physical parameters of a machine (e.g. motors' current waveform or the vibration of a pump), it is possible to detect and prevent a general failure by detecting anomalous motor behavior or the onset of a pump fault due to worn parts [21]. Furthermore, as manufacturing systems should operate continuously without performance variations, the health status of each machine must be closely monitored to reduce downtime and improve the life span of components, devices and equipments. By wirelessly monitoring temperature, oil characteristics and vibration of rolling bearings, it is possible to improve their maintenance and reduce the downtime of the manufacturing chain [22, 23].

Health monitoring encompasses the observation of physical parameters such as oxygen saturation and heart rate [24, 25] and it is useful in a large number of scenarios. The vital signal monitoring of patients in emergency scenarios allows a better tracking of the patient's health condition and a faster response to critical events [9]. Another relevant application is found in ambulatory procedures where the patient's heart rate is monitored and assumes a key importance as a diagnostic method for the detection and analysis of cardiac arrhythmia [26]. Likewise, the on-line physiological status monitoring of elderly inpatients in nursing centers can largely contribute to improve the service quality and reduce the workload of the medical staff [25]. Furthermore, health monitoring in out-of-hospital environments is also a relevant application. For example, it enables assessing the effects of a treatment at home, with the subjects living their daily life, thus providing an increased comfort to the patient and an improved medical assessment to the clinical staff [27, 28].

1.1.3 Synchronization Dependent Applications

Synchronization refers to the process of aligning the operation phase of a given set of devices so that they share a time scale origin and bounded timing difference [29]. Several applications require clock synchronization among stations. However, distributed data acquisition and network coordination are the most common. The first addresses the chal-

lenge of performing synchronous sensor sampling or actuator control in distributed nodes [30]. This is often a critical requirement of the application. For example, the acquisition of data in a given area through a set of geographically separated vibration sensors must be performed within a strict interval in order to guarantee the validity of the collected data [31, 32].

Network coordination was the focus of large research efforts, mainly in what concerns the synchronization of wireless sensor networks [33]. In this regard, synchronization has been studied in the scope of improving energy consumption, dependability, network coordination or accuracy for time based applications. Energy consumption can be highly decreased in applications requiring periodic monitoring services. This can be achieved by guaranteeing the synchronization of the stations and allowing turning on the radio transceivers only during the periods in which they are supposed to communicate with each other [34]. Furthermore, dependability can be improved by ensuring the synchronization of the stations participating in the network. For example, by adopting the time-triggered communication paradigm, communication activities are "scheduled according to a predefined, periodic scheme" that "simplifies system verification and diagnosis" [35].

Synchronization is also important for the network coordination of time slotted multiple access communication technologies. It provides the basis for guaranteeing the global consistency of a network, enabling power management optimization, dynamic frequency hopping and the implementation of time-slotted medium access control (MAC) protocols [36]. Moreover, it also improves the communications' timeliness and bandwidth usage, since it allows to significantly reduce the amount of time spent in contention for medium access. For example, in RSSI based localization applications using the IEEE 802.15.4 technology, synchronization can reduce the round trip delay by one order of magnitude in some scenarios [6].

Finally, some applications are highly dependent on synchronization to provide their functionality with the proper quality. One example is the localization based on the Time Difference of Arrival (TDOA) [37]. In this scenario, a control station polls the tags being tracked in a round robin fashion and, in response, they sent a packet which is received by multiple base stations placed in different locations. Provided that each packet travels a different distance from the tags to the base station and that they are synchronized with each other, it is possible to determine the relative and absolute position of the tag based on the instants of arrival. Hence, the better the synchronization the better the position estimation. Accuracies of up to 10 millimeters can be supported [38].

1.1.4 Requirements

Despite targeting different goals, the addressed localization, monitoring and synchronization applications share the use of wireless technologies to communicate data among stations. Hence, a comparative analysis focused on relevant communication parameters [39] can be drawn to enhance the understanding of their relative importance in each application. The parameters of the wireless communications considered for this analysis are: *dependability*, *scalability*, *goodput*, *timeliness*, *autonomy* and *range*.

The *dependability* of a communication protocol is its ability to avoid failures affecting the normal operation beyond what it is acceptable. This parameter encompasses four main dimensions [40]: availability, reliability, safety and security. Availability is the proportion of time in which the network is operating correctly. Reliability is "the duration or probability of failure-free performance under stated conditions" [41]. Safety is concerned with "the freedom from those conditions that can cause death, injury, occupational illness, or damage to or loss of equipment or property, or damage to the environment" [42]. Security is related to "the concepts, techniques, technical measures, and administrative measures used to protect assets from deliberate or inadvertent unauthorized acquisition, damage, disclosure, manipulation, modification, loss, or use" [43]. In other words, security refers to the "prevention of unauthorized access and/or handling of information" [44]. The requirements arising from these four dimensions will be holistically represented by the *dependability* parameter, which will be classified in one of three levels: "Low", "Medium" or "High", according to the stringency of the requirements. For example, an application with strict requirements concerning the communications' dependability, in either one or more dimensions, will be classified with the level "High".

Scalability is the ability to grow or shrink the size of the network supporting an application. In this analysis, scalability will be characterized by the range of stations commonly used for each application. This characterization will provide a measure of the growth, which might be required for each given application. *Goodput* refers to the number of useful bits per unit of time. This parameter will be described by the range of required bits per second (bps) in each application. *Timeliness* encompasses latency and jitter. However, in this analysis, timeliness will be represented by the communication latency, since the former is typically one order of magnitude greater than the latter [1]. *Autonomy* corresponds the time span, in days, of continuous operation without battery replacement, when applicable, and *range* indicates the distance (in meters) between transmitter and receiver that must be supported for each application. Table 1.1 summarizes the requirements of the above

Table 1.1: Typical wireless communication requirements

		Applications		
Parameter	**Quantification**	***Localization***	***Monitoring***	***Synchronization***
Dependability	Low/Moderate/High	Moderate/High	Moderate/High	High
Scalability	# of stations	[1, 1k]	[1, 10k]	[15, 10k]
Goodput	Bits per second	[10, 100k]	[0.1, 600k]	< 64
Timeliness	Seconds	[0.1, 10] $\times 10^{-3}$	[20 $\times 10^{-6}$, 360]	[72 $\times 10^{-12}$, 20 $\times 10^{-6}$]
Autonomy	D/W/M/Y*	[1W, 2Y]	[3D, 2Y]	[10D, 1Y]
Range	Meters	[1, 100]	[5, 300]	[1, 100]

* **D**ays/**W**eeks/**M**onths/**Y**ears

mentioned applications.

Localization Applications

Localization applications are characterized by a communication level of dependability ranging from moderate to high. In disaster response scenarios, for example, the employed communication technology supporting the localization must be highly reliable and ensure proper security in order to guarantee the availability of communications and the safety of people. This is required due to the critical nature of the application, i.e., the operations' command is highly dependent on both the rescuer and victim's locations to perform an adequate tactical response [9], for example. In less critical applications such as the tracking of visitors in museums or patients in medical facilities, the communications' dependability is also important, but to a smaller degree. In these scenarios, the localization application is usually deployed with a parallel access control mechanism, which allows less strict requirements regarding the reliability and availability of the localization application and, consequently, of the communication technology that supports it. Likewise, the safety requirements of such applications are usually less tighter since human lifes are not in danger.

Localization applications are quite heterogeneous in what concerns their scalability. For example, context aware applications enabled by the localization of people at home [7] usually require a small number of tags to cover all the usual inhabitants of a dwelling (typically from 1 to 10). Moreover, firefighting applications [11] have a high variation regarding the number of tags which are required to be actively being tracked in a given moment. This is motivated by the fact that different disaster scenarios pose different localization requirements in terms of responders and equipment. Resource management localization applications, such as the ones found in warehouses [13], are usually characterized for requiring

the tracking of a large number of assets (typically some hundreds).

The goodput required for localization applications ranges from a few tens of bits per second in highly sporadic localization applications (e.g., the tracking of large equipments in warehouses) to some tens of kilobits per second in more dynamic localization scenarios. For example, tracking people in a museum room requires a goodput of, approximately, 61 kbps when performing the fine-grained localization of 80 tags with a localization cycle of 1 second in a room encompassing 8 reference nodes [6]. In another example, firefighter localization requires the transmission of "beacons" with a frequency of 40 Hz [11], leading to a goodput requirement of 2560 bps, if only a byte is required to be transmitted in each "beacon".

In the context of localization applications, timeliness is typically associated with the cycle period. For instance, localization in disaster response localization scenarios can operate with cycles of up to 2 seconds [11]. Assuming a typical combat force of 10 teams, each one encompassing two firefighters, the localization of any tag should take a maximum of 100 milliseconds. Therefore, assuming that the communication delay should be one order of magnitude smaller than the localization of a single tag, the maximum delay should be bounded to 10 milliseconds.

In fine-grained tracking localization scenarios such as tracking visitors in museum exhibitions [6], the localization frequency is similar. However, the number of visitors can be considerably higher, thus shortening (in proportion) the maximum delay allowed per tag. Furthermore, in scenarios where other time critical applications are enabled by localization - as robot navigation [15], for example - the timeliness requirements become stricter, provided that localization can have a significant impact on the control algorithm, which can lead to financial loss or the injury of people. Besides the accuracy of the localization estimate, the rate at which the estimates are obtained is instrumental in supporting the avoidance of physical obstacles and a fluid navigation. Hence, considering a control loop with a period of 100 milliseconds [45], a network of anchor points with 8 nodes and the requirement of 10 RSSI "signatures" per localization round [6], the delay should be smaller than 125 microseconds.

Regarding autonomy, localization applications are considerably heterogeneous. Some applications have minor autonomy requirements concerning the communications which enable the localization. One example is the localization aided navigation in robots [15]. In this case, the amount of power employed in the communications that support the position tracking process is insignificant when compared to the power used to drive the robots'

motors, for example. In more demanding applications such as tracking workers in harsh environments [14] or victims, responders and equipments in disaster scenarios [9], the tags' autonomy should be of, at least, one week. In further demanding autonomy scenarios, such as resource management and logistics localization applications [13], tags are required to operate without maintenance for several years.

The communication range required for supporting localization applications is typically short. This is motivated mainly by the fact that this process takes advantage of the power attenuation associated to the signal propagation, which is used to estimate the distance between transmitter and receiver. Therefore, working with a smaller communication range yields a smaller localization error. Close quarters localization [14] or part localization in warehouses [13] requires very short communication ranges, typically around 1 meter. For coarse grained localization such as the tracking of firefighters in disaster scenarios [11] or localization enabled robot navigation for autonomous lawn mowing or fertilizing of large areas [15], communications can be supported on technologies with higher ranges. One popular technology being adopted is the IEEE 802.15.4 standard [46], which has a nominal range of 100 meters in open spaces.

Monitoring Applications

Monitoring applications are highly heterogeneous concerning dependability. For example, health monitoring applications [24, 25], besides requiring a high level of reliability, have strict constraints regarding the security of the information due to its private nature. Furthermore, in scenarios where the health monitoring can have a safety impact, e.g., heart rate monitoring in ambulatory procedures for the detection of cardiac arrhythmia [26], the required dependability is also high. Albeit less demanding regarding the security dimension as a consequence of its confined use, the monitoring of industrial processes [19, 20] and of machine health condition [21, 22, 23] also requires a high level of reliability. These applications can have significant financial losses when, for example, a given process fails to operate accordingly or when a part failure is not detected on time, which can result from the communication's lack of reliability. Environmental monitoring is characterized by typically having low security requirements concerning the information and moderate to high reliability requirements. There are several reasons supporting this fact. First, the transmitted information is typically not confidential and, second, the dynamics of the monitored variables are very limited (temperature, humidity and turbidity, for example). This means that, even if some data packets are lost, the environmental control algorithm

is not dramatically affected. By way of illustration, the control of an HVAC system in a warehouse [13] is capable of coping with packet loss as long as it does not affect the environmental variables being monitored/controlled. On the monitoring of environmental variables with potential higher dynamics such as the turbidity or the pollutant levels in water quality monitoring [17], the reliability of the communication technology must be high to avoid the distribution of contaminated water.

As in localization, monitoring applications have a broad range of scalability requirements. On one end, applications such as the environmental monitoring of water [17] are typically characterized by small networks (usually ranging between 1 and 10 stations). At the other end, industrial processes [19, 20], machine [21, 22, 23] and health monitoring applications [24, 25] can encompass tens, hundreds or even thousands of stations in the same network. Hence, the required scalability is highly dependent on the specific nature of the monitoring application.

The goodput necessary for supporting monitoring applications ranges from a few hundreds of bits per second in ambient monitoring applications to some hundreds of kilobits per second, for instance, in electromyography signal monitoring [47]. The goodput required for a food transport monitoring application is 400 bps, resulting from the requirement of tracking its acceleration, temperature and humidity [18]. The wireless monitoring of industrial processes includes variables such as, for example, pressure and flow. In this case, the required goodput is rather small, typically 8 bps for a refresh period of 2 seconds [19]. The temperature monitoring in plants requires a goodput of 80 bps [20].

Healthcare monitoring applications have different goodput requirements depending on the signals being acquired. To give an example, the goodput required for monitoring body temperature, systolic blood pressure, diastolic blood pressure and heart rate is of 32 kbps [25]. Monitoring supported on a three channel electrocardiogram, two channel skin impedance measurement and three dimensional acceleration information requires a goodput of 31.744 kbps [26], assuming a data representation length of 16 bits per sample.

Machine condition monitoring requires high speed data acquisition. For example, the current and vibration sampling frequency of a motor running at a speed of 2850 RPM should not be smaller than 1 kHz and 5 kHz, respectively. Hence, current and vibration monitoring requires a communication goodput of 8 kbps and 120 kbps [21], respectively, assuming 8 bit long samples. In a similar application, the temperature monitoring of rolling bearings requires a goodput of, approximately, 0.16 bps [23].

Timeliness in the context of monitoring applications is highly coupled with the com-

munications' delay (and jitter), which can range from some hundreds of microseconds, in close control loops [1], to some seconds in highly sporadic monitoring applications. To illustrate, health monitoring requires sampling periods of 3.9, 1.9 and 15.6 milliseconds for electrocardiogram, skin impedance and 3-D accelerometer signals [26], respectively. An estimate of the maximum delay and jitter can be obtained by assuming that the delay is one order of magnitude smaller than the sampling period and that the jitter is one order of magnitude smaller than the delay [1]. However, there are other vital signals whose timeliness requirements are more relaxed. For example, the monitoring of body temperature, heart rate and blood pressure (systolic and diastolic) require sampling periods of 5 seconds, 10 seconds and 1 hour [25], respectively.

Machine condition monitoring is moderately demanding regarding timeliness. Although it can be characterized by short sampling cycles of, for example, 200 microseconds for vibration signals [21], communications are not required to follow the same transmission period. If the the machine health variable being monitored is temperature, the sampling cycle becomes significantly more spaced in time, for example, 9 minutes between two consecutive measurements [23].

In food transport monitoring, the freshness and quality of strawberries can be monitored by sampling the acceleration, temperature and humidity during their transport with a period of 100 milliseconds [18]. The monitoring of industrial processes is characterized by sampling cycles of 2 seconds [19], while the temperature monitoring in the plant may adopt sampling periods of 200 milliseconds [20]. Likewise, in other scenarios, the monitoring of very slow environment variables such as PH, temperature and turbidity have even more relaxed timeliness requirements since the sampling period can be in the range of hours [16].

Monitoring applications also have heterogeneous requirements regarding autonomy. For example, application domains such as factory automation may encompass sensors and actuators installed in fixed locations with access to mains power [1] or in moving pieces [21] where the use of mains power is rather prohibitive. Industrial process monitoring [19] cannot cope with short maintenance cycles, given that they may affect the plant productivity. The same occurs in machine health monitoring. Hence, such applications should cope with maintenance cycles of more than one year. For instance, the replacement of batteries every 450 days is acceptable in the process industry [23].

Several other monitoring applications require high levels of autonomy. For example, due to the difficult access and lack of mains power sources, water monitoring has tight autonomy requirements [16]. Furthermore, food condition monitoring must be able to

cope with transport journeys of several days [18]. Regarding vital signal monitoring, different applications have different requirements. For example, the support of mobile on-line physiological status monitoring requires an autonomy of more than one month [25], while ambulatory electrocardiogram monitoring merely requires an autonomy of more than 3 days [26].

The communication range required in monitoring applications is also highly dependent of the application. In industrial settings, monitoring typically requires ranges up to 30 meters for one hop communication and around 300 meters for multi-hop communications such as those of industrial process monitoring [19]. In medical monitoring applications, for example, the continuous monitoring of fetal heart rate and intrauterine pressure requires a typical range of up to 20 meters [48]. On-line physiological status monitoring is commonly characterized by ranges from 50 (indoor) to 150 meters (outdoor) [25]. In a different monitoring application scenario, such as the transportation of food in large containers or trucks, the required range is around 5 meters [18], which is the maximum distance between any sensor in the container and the communications' gateway.

Synchronization Dependent Applications

Although wireless sensor network synchronization protocols are designed to "cope with unreliable network transmissions and unbounded message latencies" [33], they are vulnerable, to some degree, to these effects. Hence, to support a faster convergence [36] and accuracy, synchronization applications require a high level of dependability.

The scalability requirement of synchronization applications is highly heterogeneous. In synchronization applications supporting localization services, the size of each network ranges typically from 15 to 100 stations [38]. In distributed coordination application employing, for example, the Clock Sampling Mutual Network Synchronization (CS-MNS) method [36], network sizes are typically between 100 and 500 stations. In seismic activity surveying the number of geophones deployed can reach several thousands, when the area to be covered is very wide (several square kilometers) [31], for example.

Clock synchronization algorithms for wireless sensor networks can be classified in several categories, for example, *internal* or *external* [33]. In the first case, a global time reference is not available and, therefore, each station tries to "minimize the maximum difference between the readings of local clocks" [33] of the network. In this case, there is no specific message transmission for communicating the global clock reference and, hence, no goodput is required for supporting synchronization. Conversely, in *external* synchroniza-

tion algorithms, there is a global time base to which all stations seek to adjust. This is usually achieved by transmitting messages carrying the global clock reference information and allowing stations to synchronize their clocks accordingly. The period required for these messages is such that it guarantees a given synchronization accuracy. Hence, it is dependent on the stations' clock deviation. For example, the TPSN synchronization protocol [49] runs on Berkeley motes, whose clock can drift 40 microseconds per second apart [36]. For a synchronization accuracy of 100 microseconds the period of the TPSN should be of 1.25 seconds. Assuming the transmission of timestamps according to the IEEE 1588 standard, in which the time value is represented by a 80 bit number, the required goodput would be of 64 bps.

The synchronization timeliness can be associated to its jitter (and accuracy). For example, in fine grained localization applications the (RMS) jitter is of 7.2 picoseconds [50]. Furthermore, the precision of clock synchronization algorithms has been reported to be of a few tens of microseconds [51, 36]. In synchronous sensor sampling or actuator control synchronization accuracies are of under 100 microseconds or, under ideal conditions, under 1 microsecond [30]. Wireless seismic data acquisition systems require a (sampling) synchronization accuracy of 2 milliseconds [32]. Other types of synchronous sampling applications such as, for example, humidity monitoring are characterized by synchronization accuracies of 1 millisecond [34]. A timeliness estimate can be obtained by assuming, for simplicity, that the accuracy represents the maximum jitter. Hence, as defined before, the timeliness is one order of magnitude higher.

One of the key benefits of synchronization is its potential to increase the autonomy of battery powered devices. For example, in [35], an improvement by a factor of three is reported in a comparison between two scenarios: duty-cycle of 100 percent and seven percent. Another example is wireless seismic data acquisition applications [31], where synchronization allows reducing maintenance costs by increasing the networks' lifetime, which is important in long time deployment scenarios. In general, the autonomy required for synchronization applications is tightly coupled with the specific application being supported. For example, in medical environments, it should sustain the continuous operation of devices between 10 and 27 days [38].

The communication range required for synchronization applications is commonly short, typically from 1 to 100 meters [38], as the stations being synchronized are usually not too distant from each other. However, higher communication ranges have usually some impact on the synchronization accuracy, since they result in less hops and, therefore, less

communication delays [35] or in a faster synchronization convergence [36].

1.2 Motivation

The previous subsection provided an overview of three representative application categories. These applications were selected due to their adoption of personal area network technologies and operation in open spaces, where other technologies may contend for the medium. The key point of this study is the broad heterogeneity of requirements, even within the same application domain, which poses demanding challenges regarding the flexibility of the communication technologies and protocols supporting them. As the required dependability is commonly high, the communication technology should not be vulnerable to (un)intended noise and must be able to ensure, when required, the privacy of the messages being transmitted. Although the goodput needed for localization, monitoring and synchronization applications is generally small and the timeliness is highly dependent on the specific application, many of such applications have highly constrained timeliness requirements. Furthermore, the enabling communication technology should be characterized by using low power transmissions in order to support high autonomy applications with communication ranges between 1 and 100 meters.

1.2.1 Problem Statement

The support of real-time communications over license-free bands in open environments, encompassing multiple real-time stations with an unknown number of unconstrained stations, is a challenging task. Generally, the support of real-time medium access is achieved by a strict timing control of all communicating stations (real-time and non real-time). However, in open communication environments, the traffic generated by uncontrolled "alien" stations cannot be avoided by existing medium access protocols.

Many applications require the flexibility provided by wireless communications for sensor and actuator networking. Existing protocols that operate in license-free bands are only able to support reliable and timely communications in environments where bandwidth is still available to be reused (e.g., a free channel is accessible). In general, communication protocols assume that the medium has some free bandwidth or that stations make a fair use of the available bandwidth. Although these assumptions are valid for a wide set of applications, those with strict dependability and real-time requirements cannot cope with uncertainties regarding the availability of the medium. Hence, for these applica-

tions, a mechanism allowing to enforce a deterministic medium access for real-time stations is required. This mechanism should operate in open environments in scenarios where other technologies supporting non-critical applications may simultaneously contend for the medium. For example, the health monitoring of a newborn in an hospital, supported on a low-power communication technology (e.g., IEEE 802.15.4) should never be affected by nearby multimedia game streaming over Wi-Fi. Although the game streaming application requires real-time communication support, it is not critical.

The main hypothesis of this dissertation is that the use of a specific traffic separation mechanism, at the Medium Access Control (MAC) level, is the underlying foundation of a real-time communication protocol. In other words, the prioritization of privileged traffic at the MAC level is the enabling key for implementing a real-time wireless communication framework able to operate in "open environments".

The starting point of this work is the characterization of the coexistence between personal area and local area network technologies. This is motivated by the need to characterize the impact that a given technology has on others, which operate in the same frequency band and in overlapping (or near) channels. Afterwards, a novel traffic separation mechanism is proposed. This mechanism enables the prioritization of privileged traffic, allowing the coexistence of uncontrolled stations with real-time stations in a shared space. Building on this mechanism, a complete framework is designed, implemented and validated to support real-time communications in "open environments".

1.2.2 Contributions

This dissertation reports the work conducted to validate the defined hypothesis. In this scope, several studies were performed and multiple system implementations were realized and documented. The main contributions of this dissertation are summarized in the following list:

- A review of the key wireless communication requirements demanded by localization, monitoring and synchronization applications operating in the 2.4 GHz ISM band;
- A state-of-the-art and analysis of the low-power real-time communication protocols operating in the 2.4 GHz ISM band;
- A state-of-the-art and analysis of the coexistence of wireless communication technologies operating in the popular 2.4 GHz ISM band;

- The definition, implementation and validation of a novel MAC technique named *bandjacking*, allowing a deterministic wireless channel access in open environments;
- The definition, implementation and validation of the *Wireless Flexible Time-Triggered* (WFTT) protocol, enabling the support of protected real-time and best-effort communications in open environments;
- The design and implementation of an interference auditing tool named Contention-bAsed nOise Sequencer (CAOS). This tool was used to simulate environments with different levels of Wi-Fi noise, allowing to assess their impact on the WFTT communications;
- The design and implementation of a timeliness measurement tool named WIreless Timeliness Assessment System (WITAS). This tool was used to measure the delay and jitter associated to the communication of messages (packets) between stations in a WFTT network.

1.3 Outline of the Dissertation

An introduction to the relevant applications requiring wireless real-time support was provided in this chapter with focus on their communication requirements. Following, the main claims of this dissertation were presented together with the key contributions. The remainder of this section outlines the work conducted to support the stated thesis.

Chapter 2 begins by providing an overview of the main technologies currently being utilized for supporting wireless real-time protocols, with emphasis on open, low-power standards. The analysis of these technologies is focused on their features, architecture, operation and, especially, on their supported medium access mechanisms. Afterwards, a review of selected wireless real-time protocols using low-power communications for the 2.4 GHz ISM band is provided. This review encompasses both contention-based and contention-free protocols. An overview of the coexistence issues in the 2.4 GHz Industrial, Scientific and Medical (ISM) band concludes the chapter. This overview analyzes the impact that wireless technologies can have on each other.

Chapter 3 presents the *bandjacking* technique, which enforces traffic separation in open contention-based environments. This chapter begins by introducing the concept of *black-burst* and, afterwards, describes the *bandjacking* technique with emphasis on its

architecture and operation. Two physical implementations of a programmable interference synthesizer are documented and preliminarily evaluated. A testbed devised to assess the performance of the *bandjacking* technique is also described. The chapter concludes with a presentation of the collected *bandjacking* performance results and their discussion.

Chapter 4 is the "heart" of this dissertation and it is solely devoted to the proposal of the Wireless Flexible Time Triggered protocol. In this chapter an overview of the FTT paradigm is provided. Then, the WFTT protocol is presented in detail, with emphasis on the adopted design principles, architecture and operation. Furthermore, an analytical study of its timing constraints is provided focusing on its implementation feasibility and timeliness.

Chapter 5 describes the framework implementation supporting the WFTT protocol with emphasis on the architecture and operation of the developed wireless devices. Hence, the hardware and software architectures of both WFTT types of devices (master and slave) is presented, providing insights on the adopted options. Given their different nature, the operation of the master and slave devices is analyzed separately. Because the developed WFTT slaves can operate in three different modes, each one is analyzed in respect to its intended purpose.

Chapter 6 deals with the WFTT performance assessment. In this chapter, the methodology used to evaluate the WFTT protocol is characterized with focus on the devised testbed and on the timing parameters of interest. Although tools such as the WITAS and the CAOS were instrumental in the implementation of the WFTT testbed, they are characterized in Appendices A and B, respectively, due to their specificity and length. The performance results obtained using the testbed are documented and throughly discussed in this chapter.

Chapter 7 presents a summary and discussion of the contributions of this dissertation and suggests a few lines of future research that seem promising.

"If I have seen further it is only by standing on the shoulders of giants"

Sir Isaac Newton (1643 - 1727)

2 Background

The emergence of wireless low-range communication standards such as, for example, the ones of the IEEE 802.15.x family significantly contributed to the widespread adoption of low-power technologies. Building on these technologies, new protocols envisaging real-time applications have been devised. A review of the enabling low-power technologies, the real-time protocols built upon them and their coexistence with interfering technologies in the same band is provided in the following sections. The study of low-power technologies addresses the most popular protocols operating in the 2.4 GHz ISM band, namely the Bluetooth and the IEEE 802.15.4 technologies. However, due to their emergence, the ANT and nanoNET technologies are also analyzed in a section named "Emerging Technologies".

The study of the real-time protocols using these technologies is focused on factory automation applications, due to their more demanding timeliness and reliability requirements. The study provides a generic introduction to each protocol followed by a brief analysis of its operation targeting the adopted MAC mechanisms. A discussion engaged on the performance and limitations of the protocol finishes this analysis. Because these real-time protocols are built upon vulnerable technologies, an analysis of the IEEE 802.15.4 coexistence with key standard protocols operating in the 2.4 GHz ISM band (IEEE 802.15.4, IEEE 802.11 and Bluetooth) is conducted by reviewing the relevant literature.

2.1 Wireless Low-Power Technologies

During the last few years, wireless low-power communications experienced an increasing traction as a result of the higher demand posed by power constrained based applications. Such applications span over many domains ranging from home automation to health-care. The availability and broad adoption of wireless communication standards such as Bluetooth

[52] and ZigBee [53] was one of the main driving factors behind their success. Backed by an array of important companies, these technologies became massified, allowing a steep drop in their production costs and, as a consequence, an increasingly attractive cost/benefit relationship. Another key factor contributing for this success is their operation over the 2.4 GHz ISM license-free band, which is supported worldwide. Hence, the integration on consumer electronics is appealing, since the deployment of such products is facilitated when compared to wireless technologies operating in bands that are not universally supported. Nevertheless, alternative technologies such as *Z-Wave* [54] and *EnOcean* [55], which operate in the 868 MHz Band in Europe (908 MHz in the US), are also gaining some traction due to their potential for applications with low-power requirements. For example, the Z-Wave protocol focuses on WSNs for lighting, appliance control, access control and HVAC. It employs a source-routing protocol to communicate messages in a mesh network. Although it was originally developed with a data rate of just 9.6 kbps, an extension was later added allowing rates of 40 Kbps. The EnOcean technology was originally designed targeting ultra low-power communications that could be enabled by energy harvesting mechanisms. Hence, it is simpler and less rich in features. For example, it is not able to support message acknowledgement and carrier sense mechanisms.

In the following subsections an overview of the most widespread wireless short-range technologies operating in the 2.4 GHz ISM Band is provided.

2.1.1 Bluetooth

Bluetooth [52] is an open standard designed for *ad hoc* short-range wireless networking that is promoted by the Bluetooth Special Interest Group (SIG) [56]. The technology was originally developed envisaging cable replacement. However, it has evolved to be the *de facto* standard for short range communication between mobile devices. The first "solid" specification was released in February 2001 and named Bluetooth 1.1 [57]. Later, in March 2002, this release was adopted by the IEEE 802.15 working group as the IEEE 802.15.1 standard [58]. The Bluetooth 1.2 version [59] was released in November 2003 to enhance the co-existence with other wireless technologies in the 2.4 GHz ISM band and to improve audio communication. These changes were reflected in the IEEE 802.15.1 (2002) standard and resulted in an update to the standard published in June 2005 [60].

To cope with the increasing demand for higher data transfer speeds, the Bluetooth SIG launched the Bluetooth Core Specification Version 2.0 + Enhanced Data Rate (EDR) [61] in November 2004. Besides the increased data rate, this specification brought im-

provements such as a lower power consumption and a simplified multi-link mechanism, maintaining backward compatibility with the previous versions. In July 2007, the Bluetooth Specification Version 2.1 [62] added secure simple pairing to improve security and usability.

The continuous market demand for higher speed communications resulted in the publication of the Bluetooth Specification Version 3.0 + High Speed (HS) [63], which increased the maximum data rate by adopting an hybrid approach based on the MAC and PHY layers of the IEEE 802.11 standard [64]. More recently, since June 2010, the Bluetooth SIG adopted the Version 4.0 specification [52], which adds the Bluetooth Low Energy (BLE) to the set of supported protocols (Bluetooth classical and HS).

Overview

The Bluetooth technology operates in the 2.4 GHz Industrial, Scientific and Medical (ISM) band and employs a Frequency Hopping Spread Spectrum (FHSS) technique through 79 evenly spaced 1 MHz channels, ranging from 2.402 MHz to 2.480 MHz, in order to lower the signal fading and interference associated to the wireless channel. The hop rate is 1600 hops per second so that each hop is separated from the next by a 625 microseconds time window. The Bluetooth channel is a specific pseudo-random hopping pattern that is derived from the network master's clock and address. Provided that the channel is divided in time slots, the Time Division Duplex (TDD) scheme is used for transmissions. Packets can occupy one, three or five slots. Slots can also be reserved for synchronous transmissions, e.g., voice data.

The Bluetooth technology is classified in one of the three possible power classes: 1, 2 or 3. Class 1 devices are designed for long range (100 meters) communications and transmit signals with a maximum output power of 100 mW, whereas class 2 and 3 devices aim at supporting short-range communications between 10 centimeters (class 3, 1 mW) and 10 meters (class 2, 2.5 mW).

The "classical" Bluetooth Core Specification Versions 1.1 and 1.2 define a radio employing a Gaussian Frequency-Shift Keying (GFSK) modulation scheme, which allows a symbol transfer rate of 1 Mbps (effective data rate of 723 kbps over-the-air) designated "Basic Rate" (BR). Versions 2.0 and 2.1, although maintaining the compatibility with the Bluetooth BR versions, add two modulation schemes: $\pi/4$-rotated Differential-encoded Quaternary Phase-Shift Keying ($\pi/4$-DQPSK) and Differential-encoded 8-ary Phase-Shift Keying (8DPSK). These "Enhanced Data Rate" (EDR) versions support symbol transfer

rates of 2 Mbps and 3 Mbps, respectively. However, to maintain compatibility, only the payload section is encoded using the "enhanced" modulation schemes.

Despite of the data rate improvement introduced by the "enhanced" modulation schemes, Bluetooth was still much slower than, for example, Wi-Fi, deterring its broad adoption for transferring large files. Hence, in a effort to improve the technology's throughput, the Bluetooth SIG adopted the Version 3.0 release, supporting a significantly higher data rate of up to 25 Mbps. This is achieved by employing an alternate MAC/PHY that allows Bluetooth to cooperate with a technology supporting an higher data rate (IEEE 802.11). Therefore, the Bluetooth radio is used (initially) to establish the pairing procedure and set the security mechanisms. When a higher throughput is required, the Bluetooth module switches to the alternative IEEE 802.11 radio technology.

Besides the continuous demand for higher throughput, wireless technologies have also experienced a trend in a different direction, i.e., the demand for extremely low-power communications that can enable applications with strict autonomy requirements (several years of continuous operation). The evolution in mobile devices over the past few years has fueled the appearance of new types of applications requiring a tight integration with embedded low-power sensors such as fitness devices, watches, clothes, etc. Given the limitations of the connection-oriented operation of the Bluetooth technology, a new protocol was devised to coexist with Bluetooth and enable devices to operate in sleep mode for the most part of their lives. This protocol is named Bluetooth Low Energy (BLE) [52]. Its origin can be tracked to the early days of the IEEE 802.15.4 standard development. Throughout its evolution it took several designations, namely: Blulite, Wibree, Bluetooth Ultra-low Power, and, finally, Bluetooth Low Energy.

The BLE emerged from the need to develop a protocol that could coexist with standard Bluetooth and, more importantly, that could share the Bluetooth's radio structure to support both protocols in a dual-mode chip architecture. These chips can then be integrated by manufacturers in consumer devices such as mobile phones and tablets, allowing the support of both protocols. Low-power devices use a single mode chip that only provides the features associated to the BLE, thus enabling low-cost products supporting ultra-low power operation. The BLE radio supports a subset of 40 channels in the 2.4 GHz band, adopting a more relaxed Gaussian frequency-shift keying (GFSK) modulation scheme when compared to the Bluetooth BR/EDR and allowing increasing its operating range. In addition, the BLE protocol uses very short packets (10 to 47 octets) and allows the completion of a connection establishment plus data transfer in a period as short as 3

milliseconds.

Unless an attacker is in possession of the frequency hopping sequence, Bluetooth transmissions are relatively difficult to intercept. The Bluetooth standard defines that under normal operation the frequency-hopping pattern cannot be determined. However, during the pairing procedure and due to specific implementations of the Bluetooth standard, discoverable devices are vulnerable to attacks. For example, one of the earliest Bluetooth security threats is known as *BlueSnarf* and it allows an attacker to steal information (e.g., the phonebook) from mobile devices by exploiting a flaw in the firmware of older devices. The *National Institute of Standards and Technology* studied the Bluetooth vulnerabilities and threats [65], including *Denial of Service* (DoS), eavesdropping, *man-in-the-middle* (MITM), message modification, and resource misappropriation attacks. This study provides several countermeasures and guidelines, which allow reducing the risk of their occurrence. Nonetheless, the use of the Bluetooth technology in critical applications such as mobile health-care should be avoided due to the lack of privacy caused by the "information leakage of the pairing process, the lack of security and the resistance to interference" [66]. Nevertheless, the Bluetooth's Adaptive Frequency Hopping (AFH) mechanism will still provide a higher probability of transmission success when compared to DSSS-based technologies, given that it will automatically avoid the use of jammed frequencies.

Architecture

Two or more units sharing the same Bluetooth channel form a piconet (see Figure 2.1). The piconet is the Bluetooth's atomic network. Each piconet has one piconet master and one or more piconet slaves (a maximum of seven active slaves). However, several slaves might be attached to the network in a so-called *parked* state. These parked slaves are not active but are synchronized with the piconet master. Piconets covering common areas form a scatternet. Bluetooth units belonging to one piconet may participate in other piconets by using a Time Division Multiplex (TDM) scheme. Moreover, a piconet master may be a piconet slave in a neighbor piconet. Piconets are not frequency hop synchronized, meaning that each one has its own hop sequence. This partially avoids piconet interference.

Operation

The Bluetooth protocol supports two types of connections: point-to-point (only two Bluetooth units communicate with each other) and point-to-multipoint. In the latter form, the channel is shared with all connected Bluetooth units. In Bluetooth, two types of links

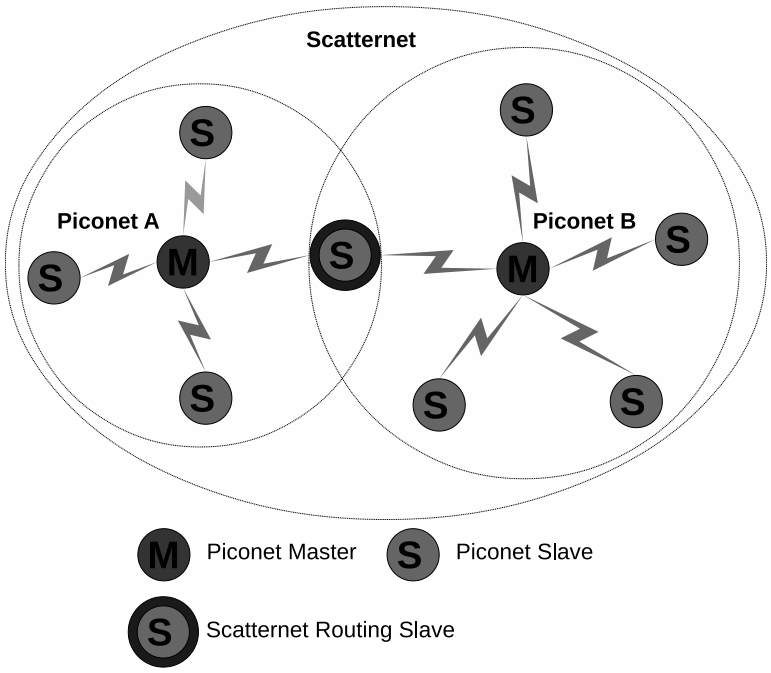


Figure 2.1: Bluetooth scatternet

are defined: Asynchronous Connectionless (ACL) and Synchronous Connection-Oriented (SCO). ACL links carry best-effort traffic and are suited for asynchronous transmissions. SCO links support periodic data transmissions at a 64 kbps rate in each direction. SCO traffic cannot be retransmitted and thus can only recover from errors by using Forward Error Correction (FEC) mechanisms. Since specification 1.2, a limited number of retransmissions is possible by using the extended SCO (eSCO) link type, which improves the quality of the link by allowing the retransmission of corrupted packets.

As pointed, the communication channel is divided in time slots, each one occupying 625 microseconds. These slots are numbered according to the piconet's master clock. The master transmits on even numbered slots and the slave on odd numbered slots. Each transmission takes place at one new hopping frequency, and a complete data packet is sent in each slot. This means that there is no frequency shift before the end of the packet transmission, even if the packet occupies more than one slot. Figure 2.2 depicts the communication process for a single slave piconet.

Point-to-multipoint communication occurs when a piconet contains more than one slave. In this scenario, and to prevent piconet members of jamming each other with simultaneous transmissions, a Time Division Duplex (TDD) scheme is used. This means that the overall throughput is multiplexed by the slaves. Figure 2.3 shows the procedure. As before, the

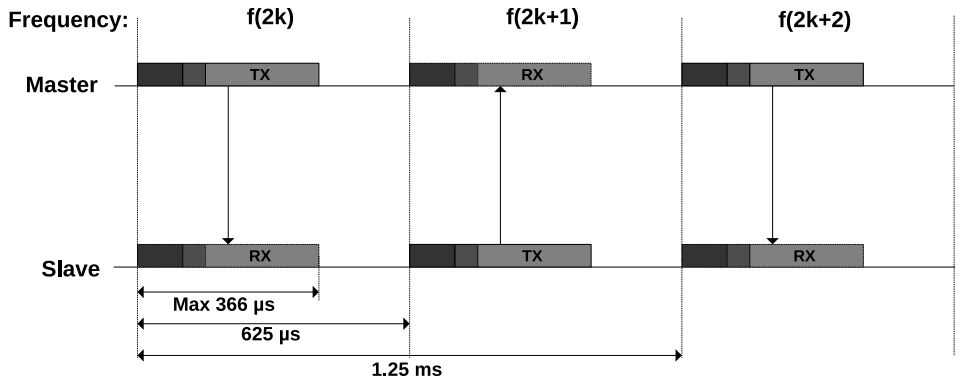


Figure 2.2: Single-slave operation

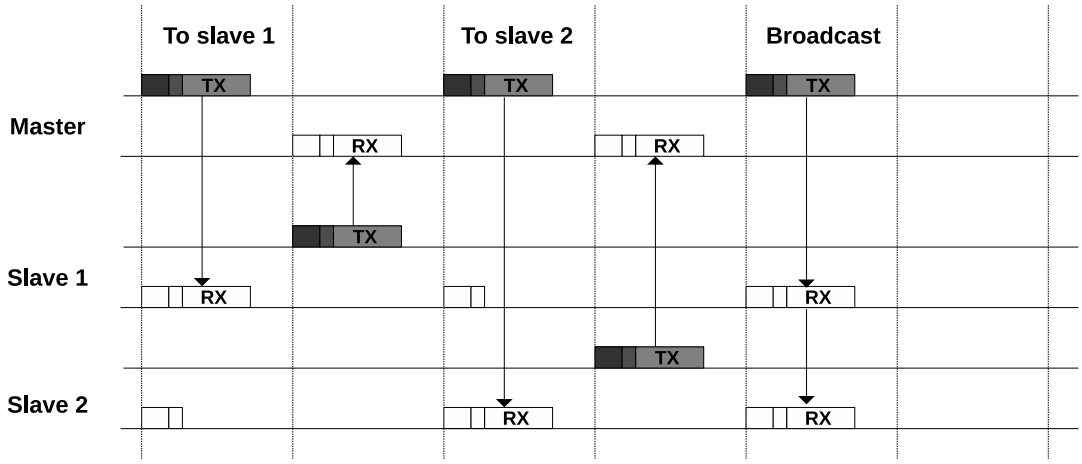


Figure 2.3: Multi-slave operation

master transmits on even-numbered slots and the slaves on odd-numbered slots. However, slaves can only transmit when directly addressed by the master in the previous time slot. In this sense, if the master sends a packet to slave 1, slave 2 will remain with its receiver turned on only while decoding the packet access code (identifying the piconet) and header (identifying the destination). Afterwards, it will realize that the packet was not addressed to him and will turn off its receiver until the next even-numbered slot.

Figure 2.3 also demonstrates that piconet slaves only communicate with the piconet master, which means that slave-to-slave data transfers are merely possible through the master or by creating a separate piconet in which they can directly communicate (one assuming the role of piconet master and the other of piconet slave). The support of multi-slot packets enables increasing the achievable data throughput. Multi-slot packets can have a duration of either three or five time slots. However, they all must be sent in a single hop

frequency.

As mentioned, the co-existence with other wireless technologies was addressed by introducing the Adaptative Frequency Hopping (AFH) mechanism. This mechanism works by reducing the number of nominal hopping channels according to their occupation by other ISM technologies such as Wi-Fi, microwave ovens, cordless telephones, etc. In this sense, the occupied channels are marked and are avoided in the hopping pattern.

Power Consumption and Availability

Motivated by the focus on low-power technologies operating in the 2.4 GHz ISM band, this subsection specifically addresses the Bluetooth BLE technology, which is available from manufacturers such as Nordic Semiconductor [67], Texas Instruments [68], CSR [69] and EM Microelectronic [70]. Table 2.1 presents a summary containing the main characteristics of the devices produced by these manufacturers.

Table 2.1: Bluetooth Low Energy commercial solutions

Parameter	nRF8001	nRF8002	CC2540	CSR1010	CSR1011	EM9301
Type	Transceiver	Transceiver	SoC	SoC	SoC	Transceiver
Package	QFN32	QFN32	QFN40	QFN32	QFN56	QFN24
Size (mm)	5 x 5	5 x 5	6 x 6	5 x 5	8 x 8	5 x 5
Supply (V)	2.3 - 3.6	2.3 - 3.6	2.0 - 3.6	1.8 - 3.6	1.8 - 3.6	0.8 - 1.8
Max. Power (dBm)	4	4	4	7.5	7.5	3
Sensitivity (dBm)	-87	-87	-93	-92.5	-92.5	-80
Max. Current (mA)	14.6	14.5	19.6	16	16	12.9
RAM (KB)	NA	NA	8	64	64	NA
ROM (KB)	NA	NA	128 or 256	64	64	NA
Sleep Current (μA)	0.5	UA	0.4	5	5	9
Price Estimate (€)	3.76	2.3	3.71	2.21	2.4	UA

NA - *Not Applicable*
UA - *Unavailable*

Nordic Semiconductor offers two ICs ($\mu Blue$) supporting the BLE protocol: nRF8001 and nRF8002. The first is a BLE transceiver designed to operate in a peripheral role connected to a host microcontroller. This device requires peak currents of 14.6 and 12.7 milliamperes to support active reception and transmission (@ 3 Volts and 0 dBm), respectively. The second is tailored to the development of low energy peripheral proximity solutions using a peak current of 14.5 milliamperes in any mode of operation (@ 3 Volts).

Texas Instruments supports the BLE technology with its system-on-chip (SoC) CC2540 IC, which is capable of running simultaneously both the application and the protocol

stack, thus dispensing the need of a host microcontroller for processing. Furthermore, the CC2540 includes the memory required for supporting both stack and application in the same device (128 to 256 KB of Flash memory and 8 KB of RAM). The device operates in active reception and transmission modes (@ 3 Volts) with current consumptions as low as 19.6 and 24 milliamperes, respectively.

CSR offers the μEnergy platform for BLE enabled applications. The CSR1010 and CSR1011 single-mode Bluetooth low energy SoCs provide the elements (RF, baseband, MCU, Bluetooth v4.0 stack) required to create a Bluetooth low energy product encompassing a user application running on a single device. The CSR1010 and CSR1011 include enough memory to support both the Bluetooth stack and the user application (64 KB of ROM and 64 KB of RAM) on the same device. Their power consumption (@ 3 Volts) is of 16 milliamperes, in active mode, and less than 5 microamperes, in sleep mode.

Finally, EM Microelectronic has introduced the EM9301 transceiver in 2012. As Nordic transceivers, this device requires a host microcontroller to run the BLE profiles and the user application. The EM9301 transceiver is intended for ultra low-power applications. Hence, it requires just 12.9 milliamperes in active mode (@ 1.2 Volts) and 9 microamperes in sleep mode. Furthermore, the DCDC version operates with voltages as low as 0.8 Volts, which makes it highly adequate for single-cell battery (1.5 Volts) operation.

2.1.2 IEEE 802.15.4

The IEEE 802.15.4 [46] is a communication standard that specifies the Media Access Control (MAC) sub-layer and physical layer for Low-Rate Wireless Personal Area Networks (LR-WPANs). The first approved version of the protocol was released in 2003 [71] and has experienced a broad support from the industry, later being adopted as the base for ZigBee [53], WirelessHART [72] and 6LoWPAN [73] technologies, among others. ZigBee is a low-power, low-rate popular protocol used in a large number of applications including home automation, industrial control and environmental sensing, just to name a few. WirelessHART is a specialized protocol that became popular in factory and process control. It allows extending existing wired Highway Addressable Remote Transducer Protocol (HART) networks with wireless connectivity. The 6LoWPAN is a wireless protocol that aims at bringing the Internet protocol connectivity to small, low-power wireless devices. In this sense, it aims at addressing every wireless device of the 6LoWPAN network using IPv6. 6LoWPAN is an acronym of "IPv6 over low-power wireless personal area networks".

A revised version of the IEEE 802.15.4 protocol was published in September 2006

[74]. This version partially addressed the requirements posed by the ZigBee specification. Besides providing clarifications to the previous version, it focuses on increasing its flexibility and security while reducing complexity. This updated version encompassed increased data rates and new modulation schemes for the 868 and 915 MHz physical layers. Furthermore, a new modulation scheme addressing the 2.4 GHz ISM band was also introduced.

In March 2007, an IEEE 802.15.4 amendment [75] was approved adding two physical layers to the standard: UWB Pulse Radio and Chirp Spread Spectrum, both aimed at decreasing power consumption, increasing the communication range and the aggregate throughput. The included UWB Pulse Radio PHY is based on the Direct Sequence UWB technology, which can support accurate ranging (up to 1 meter precision) and allow robust communications to occur even at low power levels. The Chirp Spread Spectrum adds the ability to support wireless communications to devices moving at high speeds.

The active version of the IEEE 802.15.4 standard [46] was approved in June 2011 by the IEEE Standards Association and, in August 2012, by the American National Standards Institute (ANSI). The IEEE Standards Association states that this revision addresses the extension of the market applicability (possibly regarding the 6LoWPAN protocol), the ambiguities in the previous version and the improvements "learned from implementations of IEEE Std 802.15.4-2006".

Overview

The current version of the IEEE 802.15.4 technology [46] supports a broad range of physical layers, each one defining different frequency bands and data rates. Table 2.2 organizes these bands and data rates by the associated spreading and modulation technologies, as defined in the IEEE 802.15.4 standard. Because this document specifically addresses low-power communication technologies that can be supported worldwide, it focuses on the 2.4 GHz ISM band, in particular on the IEEE 802.15.4 implementations using the Direct Sequence Spread Spectrum DSSS technique and the Offset Quadrature Phase-Shift Keying modulation method to perform packet transmissions. In this case, the IEEE 802.15.4 technology supports 16 channels with a data rate of 250 Kbps.

The IEEE 802.15.4 standard defines a minimum output power of -3 dBm in most bands. Nevertheless, some bands allow the emission of more power, depending on the type of spreading technology employed. For example, spread spectrum technologies can perform packet transmissions in the 2.4 GHz ISM band with power levels of up to 100 mW (20 dBm) of Effective Isotropic Irradiated Power (EIRP). Provided that the communication range

Table 2.2: IEEE 802.15.4 physical layers

Spreading Technology	Modulation	Bands (MHz)	Data Rate (Kb/s)
CSS	DQPSK	2400.0 - 2483.5	1000
DSSS	BPSK	868.0 - 868.6	20
		902.0 - 928.0	40
		950.0 - 956.0	20
	MPSK	779.0 - 787.0	250
	O-QPSK	779.0 - 787.0	250
		868.0 - 868.6	100
		902.0 - 928.0	250
		2400.0 - 2483.5	250
None	GFSK	950.8 - 955.8	100
PSSS	ASK	868.0 - 868.6	250
		902.0 - 928.0	250
UWB	BPM-BPSK	249.6 - 749.6 3100.0 - 4800.0 6000.0 - 10600.0	110 - 27240

is usually dependent on the transmission power and on the sensibility of the receiver, the use of power amplifiers (to increase the transmission power) and low-noise amplifiers (to increase the reception sensibility) can contribute significantly to improve the transmission range of the IEEE 802.15.4 technology. The indoor nominal communication range of standard IEEE 802.15.4 nodes transmitting with power levels of 0 dBm and 20 dBm is of 10 and 100 meters, respectively.

IEEE 802.15.4 communications are enabled with several security mechanisms, which are supported by the protocol's MAC sublayer. These mechanisms are provided in the form of services, namely: confidentiality, authenticity and replay protection. The former is related to the requirement of making the transmitted data only available to the addressed node, and no other. This service is implemented using encryption, guaranteeing that adversaries cannot recover the full message nor have access to any partial information that was encrypted. In order to meet this requirement, two identical data messages must originate two different cyphertexts. This is achieved by including a *nonce* in each encryption, i.e., an ever changing input that adds variability to the information being encrypted, thus making the encryption of identical data messages result in very different cyphertexts. The

IEEE 802.15.4 nonce is a 13-octet string including the source address (8 bytes), the frame counter (4 bytes) and the security level (1 byte) fields.

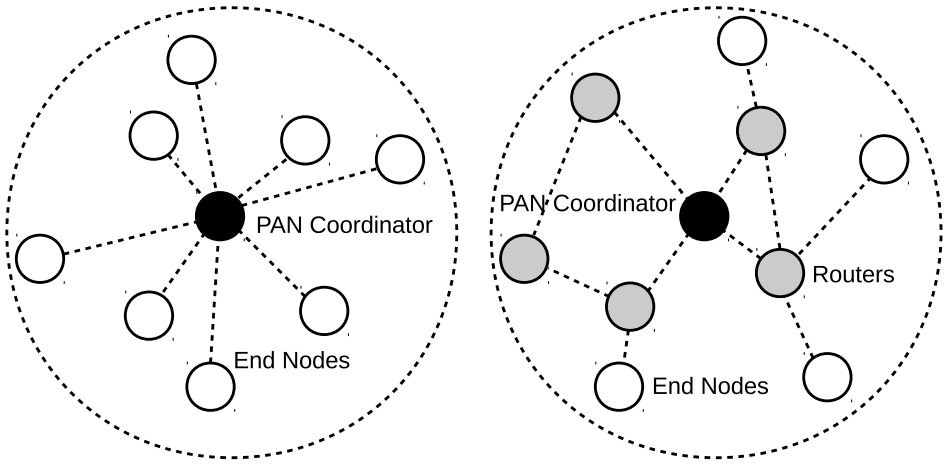
The authenticity service provides the means to ensure that the received data is the one that was effectively transmitted, i.e., that it was not modified in transit. The inclusion of a Message Integrity Code (MIC) in each exchanged packet allows verifying its authenticity and integrity. This MIC can be simply viewed as an encrypted message checksum, whose calculation requires a secret cryptographic key shared among the network's genuine nodes. In this sense, in addition to the data information itself, secured packets encompass a MIC that is computed using the packet together with the secret shared key. Secured messages are only accepted if the locally computed MIC is identical to the one embedded in the packet. Otherwise, nodes reject the forged message.

The replay protection service ensures the detection of duplicate transmissions, i.e., the occurrence of attacks involving the transmission of previously sent (legitimate) messages. Because adversaries can replicate packets with a valid MIC, receiver(s) may acknowledge them as legitimate and accept the packets. The replay protection service is implemented by including a frame counter field in the transmitted packet, which is monotonically incremented at each exchanged packet. When the receiver detects a packet with a frame counter smaller than one already received, it is rejected.

Security services are optional and, depending of the service, can be implemented using different alternative mechanisms. When enabled, security services are applied to each transmitted/received frame. The replay protection service is enabled whenever one or both of the confidentiality and authenticity services are activated. Furthermore, all frame types can be protected using these services, except the acknowledgment frame. The IEEE 802.15.4 standard defines several symmetric-key cryptographic mechanisms to support these security services. The keys used by the services are shared among peers or a group of devices and are provided by the higher layers of the network's protocol stack.

Architecture

The IEEE 802.15.4 technology supports two types of device: Full Function Device (FFD) and Reduced Function Device (RFD). The FFD is typically a full featured device that is able to perform the role of coordinator (or router), besides being able to communicate with any other devices in its radio range. The FFD can assume the special role of Personal Area Network (PAN) coordinator. The PAN coordinator is responsible for establishing and maintaining the network. The FFD, due to its broader application and



(a) Star (b) Peer-to-Peer

Figure 2.4: IEEE 802.15.4 topologies

more demanding resources (memory and CPU power), tends to be more expensive than the RFD. The RFD is a simpler device that associates with a coordinator (or router) and is only able to communicate with FFD devices, since it encompasses no routing capability. Therefore, it is used as an "end device", typically a sensor or actuator. These types of devices also differ regarding the requirements of low-power operation. Because FFDs may be required to forward messages from other nodes, they must remain actively listening for those messages. When this occurs, FFDs are commonly mains powered. Conversely, because RFDs do not require to route messages, they may remain in sleep mode for long periods of time, only waking up when there is some new information to be sent. This type of devices is more suited for applications requiring the use of batteries.

The IEEE 802.15.4 supports both star and peer-to-peer topologies, as documented in Figure 2.4(a) and 2.4(b), respectively. The star topology is developed around a central PAN coordinator that is connected directly to a set of different nodes. Although these nodes can communicate with the PAN coordinator, they are not able to perform packet transmissions among themselves, even if some are FFDs. As introduced, the peer-to-peer network is also established by the PAN coordinator. However, some FFDs devices are allowed by the PAN coordinator to act as routers, i.e., to establish direct connections among FFDs. In this sense, some devices (either FFDs or RFDs) act like "end devices", only being able to communicate with the PAN coordinator, while other FFDs (routers) can talk to each other.

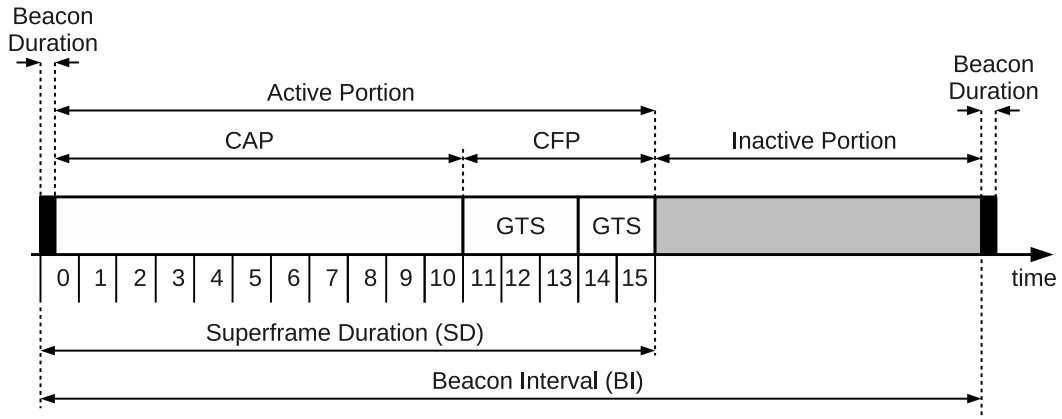


Figure 2.5: IEEE 802.15.4 superframe example

When the geographical area being covered by the network is beyond the nominal range of the IEEE 802.15.4 technologies, two topologies can be employed: cluster-tree and mesh. The former corresponds to a set of star networks interconnected by FFDs, which act as routers, forwarding packets through the network backbone. The latter topology can be implemented in a similar fashion. However, it requires higher layers of software on the nodes to be implemented in order to support the required network formation and routing services.

Operation

The MAC layer defines two operation modes: beaconed and non-beaconed (also known as beaconless). In beaconed mode, the access to the physical radio channel is ruled by a superframe structure enforced by the PAN coordinator. As documented in Figure 2.5, the superframe is bounded by the transmission of beacon frames and encompasses an active portion and an optional inactive portion. The former corresponds to the period of time in which all communications take place. The latter is used to allow entering a low-power (sleep) mode in which no communications are expected and the radios can be turned off, thus increasing the autonomy of nodes powered with batteries.

The active portion of the superframe is segmented into 16 slots and the beacon frame is transmitted at the beginning of the first slot (zero), as depicted in Figure 2.5. The Contention Access Period (CAP) follows immediately. During this period, the access to the channel is performed by all nodes using the slotted Carrier Sense Medium Access/Collision Avoidance (CSMA/CA) access mechanism and all initiated transactions must complete within the bounds of this time window. The CSMA/CA is a technique that has been

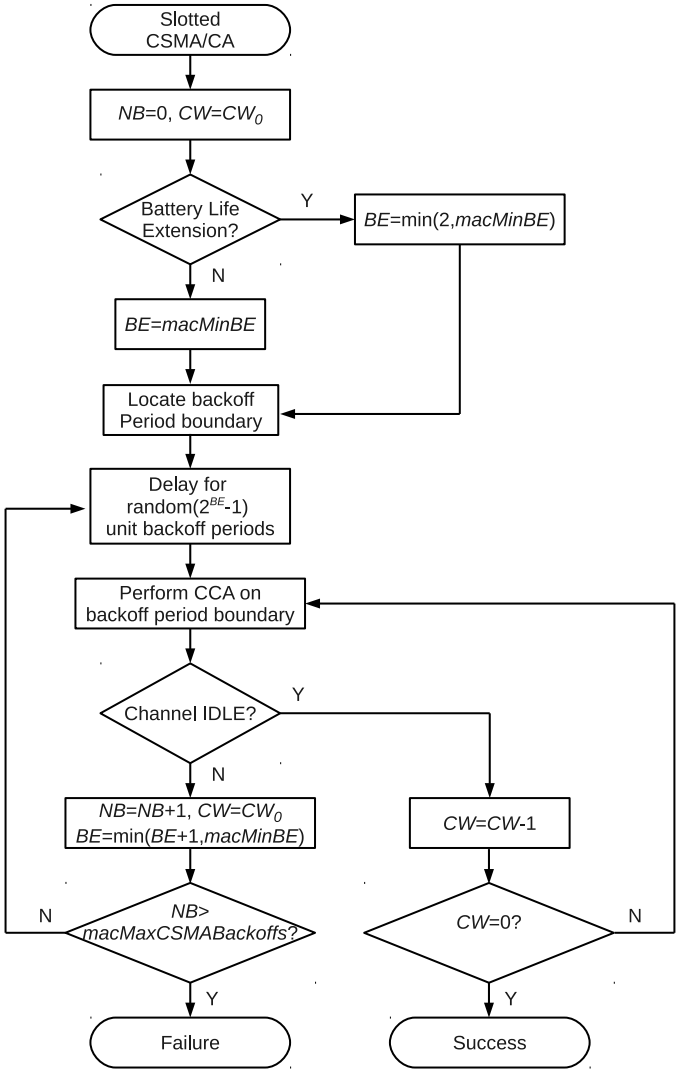


Figure 2.6: IEEE 802.15.4 slotted CSMA/CA algorithm

popularized by its "non-greedy approach" in distributing bandwidth among contending stations. The slotted CSMA/CA algorithm is represented in Figure 2.6. As shown, the first step corresponds to setting the initial values for the variables NB and CW. The former refers to the number of retries that the algorithm was required to back off in attempting to perform the transmission. The latter corresponds to the contention window length, which defines the number of required backoff periods without channel activity before a transmission can be initiated.

If the Battery Life Extension (BLE) bit is set, the Backoff Exponent (BE) is set to the minimum value between 2 and *macMinBE*. Otherwise, the BE is set simply to *macMinBE*. Afterwards, the next backoff period boundary is located and a random delay ranging from

0 to $2^{BE} - 1$ unit backoff periods is enforced. When the delay elapses, the algorithm checks whether the current CAP can accommodate the remaining operations (CCA checks, packet transmission and associated acknowledgement), proceeding to perform the Clear Channel Assessment (CCA) check if the necessary time is available, or halting until the next active portion of the subsequent superframe, otherwise. As Figure 2.6 illustrates, the CSMA/CA algorithm only allows a packet transmission to begin after performing a CW_0 number of CCA checks finding the channel idle. Alternatively, if a CCA check detects the channel busy, the algorithm assumes that a transmission is ongoing and restarts the CSMA/CA algorithm. In this sequence, NB is incremented by one unit, CW is set to CW_0 and the minimum of $BE + 1$ and *macMinBE* is assigned to BE. If NB exceeds the *macMaxCSMABackoffs* backoff cycle limit, the CSMA/CA algorithm declares a channel access failure and reports this event to the higher layers of the protocol, which manage the occurrence.

The CAP is optionally followed by a Contention-Free Period (CFP) where bandwidth can be reserved for real-time transmissions of individual nodes. Hence, when a node needs to perform a contention-free access in order to transmit a packet, it requests a Guaranteed Time Slot (GTS) indicating the duration which ensures a proper packet transmission. The PAN coordinator is, then, responsible for allowing or denying the request. If the request is accepted, the node gains exclusive access to a designated GTS. However, the packet transmission must be concluded one IFS period before the end of the assigned GTS. As introduced, when present, the inactive portion of the superframe allows the nodes participating in the network to switch to a low power mode. Besides saving energy, the inactive period can also be used to establish the interconnection between IEEE 802.15.4 clusters.

Regarding the beaconless mode of operation, the access to the medium is ruled by a simpler CSMA/CA mechanism, as represented in Figure 2.7. The main difference to the beaconed CSMA/CA algorithm is the absence of the superframe defined temporal bounds, which allows several simplifications in the algorithm. First, the random backoff countdown can begin immediately after the packet transmission has been requested by the upper layers. Second, when the random backoff expires a single CCA check is enough to allow the data packet transmission if the medium is found idle. Third, because there is no superframe, a node can perform a CCA check, transmit a data packet (if the medium is available) and receive the corresponding acknowledgement (if requested) immediately after the end of the random backoff interval.

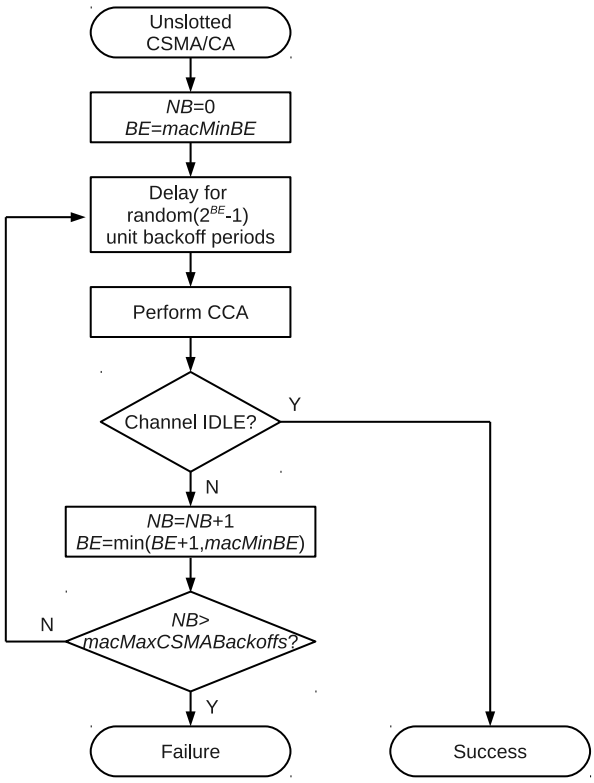


Figure 2.7: IEEE 802.15.4 unslotted CSMA/CA algorithm

A brief description of the algorithm presented in Figure 2.7 follows. As documented, BE is initialized to the value of *macMinBE*. If *macMinBE* is set to zero, the first iteration of the algorithm will run without collision avoidance. As introduced, NB is the variable registering the number of times the CSMA/CA algorithm was required to backoff while attempting the undergoing transmission. This variable is initialized to zero before each new transmission attempt. After initializing NB and BE the algorithm waits a random number of backoff periods in the range of 0 to $2^{BE} - 1$, before requesting the PHY to perform a CCA. If the channel is perceived idle, the transmission of the frame is immediately initiated. Contrarily, if the channel is found busy (e.g., the detected energy level is above a specific threshold), both NB and BE parameters are incremented by one unit, ensuring that BE is bounded by *macMaxBE*. Afterwards, if NB is smaller than *macMaxCSMABackoffs*, the algorithm returns to the state of waiting a random number of backoff periods. Otherwise, as in the beaconed operation mode, the algorithm reports a channel access failure to the higher layers of the protocol. For more detailed information about the IEEE 802.15.4 technology, please consult its specification [46].

Power Consumption and Availability

Given the broad adoption of the IEEE 802.15.4 technology, many manufacturers are providing hardware capable of supporting this protocol. In this subsection, a short summary of some products provided by the key manufacturers is provided for reference. Hence, devices from Freescale [76], Texas Instruments [68], Atmel [77] and Microchip [78] are presented and discussed in Table 2.3, which shows the most representative parameters of the selected devices. In this table, two Freescale IEEE 802.15.4 devices are documented: MC13233 and MC13202. The former is a SoC that can simultaneously integrate an application and a communication stack, due to its support of 5 KB of RAM and of 82 KB of FLASH memory. The latter is a transceiver that requires a host microcontroller to be fully functional. As documented, although their current consumption in sleep mode is close, the transceiver requires a higher current consumption (42 milliamperes) than the SoC (35 milliamperes) when performing a packet reception.

Table 2.3: IEEE 802.15.4 commercial solutions

Parameter	MC13233	MC13202	CC2538	CC2520	AT86RF231	MRF24J40
Type	SoC	Transceiver	SoC	Transceiver	Transceiver	Transceiver
Package	LGA48	QFN32	QFN56	QFN28	QFN32	QFN40
Size (mm)	7 x 7	5 x 5	8 x 8	5 x 5	5 x 5	6 x 6
Supply (V)	1.8 - 3.6	2.0 - 3.4	2.0 - 3.6	1.8 - 3.8	1.8 - 3.6	2.4 - 3.6
Max. Power (dBm)	2.3	3	7	5	3	0
Sensitivity (dBm)	-94	-92	-97	-98	-101	-95
Max. Current (mA)	35	42	34	33.6	14	23
RAM (KB)	5	NA	32	NA	NA	NA
ROM (KB)	82	NA	128, 256, 512	NA	NA	NA
Sleep Current (μA)	0.45	1	0.4	<1	0.02	2
Price Estimate (€)	5.85	2.78	6.94	3.44	2.34	2.15

NA - *Not Applicable*
UA - *Unavailable*

Texas Instruments is also represented in Table 2.3 with one SoC (CC2538) and one transceiver (CC2520). Although it adds an ARM Cortex M3 processor and memory to the basic functions of a transceiver such as the CC2520, the CC2538 can operate with a maximum current consumption (34 milliamperes @ $7dBm$) similar to the CC2520 transceiver (33.6 milliamperes @ $5dBm$), when performing the transmission of a packet. However, the latter has a broader voltage supply range and it is much less expensive. Two other transceivers were also considered, the AT86RF231 from Atmel and the MRF24J40 from

Microchip. The former, is characterized by the best sensitivity and smallest current consumptions of the group. The second, has an intermediate sensitivity and current consumption, but presents the lowest cost per unit of the considered solutions.

2.1.3 Emerging Technologies

The 2.4 GHz ISM band has become the *de facto* unlicensed band for wireless low-power communications worldwide. In this sense, Bluetooth, IEEE 802.15.4 and its derivate protocols have pushed the popularity of this region of the spectrum to levels unknown before to any short-range technology. Despite their market dominance, these technologies present some areas of improvement (e.g., the immunity to multipath fading), which can be explored to create new competing technologies. In the following section, two recent low-power technologies operating on the 2.4 GHz ISM band are presented and briefly discussed.

ANT

ANT [79] is a proprietary wireless sensor network protocol operating in the 2.4 GHz ISM band designed for ultra-low power, ease of use, efficiency and scalability. The protocol was originally developed by Dynastream Innovations Inc. (a Garmin subsidiary) to allow communications between running shoes and a wristwatch display. However, given its potential for enabling bio sensors, home sensors and industrial sensors with ultra-low-power communications, chips with built in ANT protocol were made available to third party developers. Provided the market traction for this protocol, several smartphone manufacturers are already embedding this technology in their headsets. For example, the recent Galaxy S4 and Note 3 smartphones from Samsung already have ANT+ support. The same occurs for the Xperia range of Sony smartphones (Acro S, ion, active, arc, S, etc.).

According to the OSI reference protocol stack model, the ANT protocol implements the physical, data-link, network and transport layers, including low-level security features. The application and presentation layers together with the high-level security can be added by the developers, which implement the networking applications. While the ANT protocol defines the basic networking functionality, the ANT+ protocol establishes the support required for interoperability among ANT nodes. In this sense, it is an open application layer that lays on top of the ANT stack, standardizing communications and enabling interoperability among different ANT devices (sports, wellness and lifestyle monitoring).

The ANT protocol can operate under several network topologies, namely: star, peer-

to-peer, tree and mesh, among other variations. The protocol's support for data communications is highly flexible, allowing data transfers to be scheduled in both an ad-hoc or deterministic manner. Besides these transfer modes, the ANT protocol supports a burst mode, which enables transferring large amounts of data to/from computational devices in an efficient way. A maximum of 125 channels, each one with a bandwidth of 1 MHz, is supported by the ANT protocol. Since these channels cover the 2400 to 2524 MHz region of the spectrum, before using a channel, a check to the compliance with the applicable RF emission regulations is required. The protocol employs a GFSK modulation scheme and operates with a data rate of 1 Mbps. The medium access is based on a TDM technique in which nodes transmit in designated time slots.

Table 2.4: ANT commercial solutions

Parameter	nRF24AP1	CC257x	ANTAP281M4IB
Type	Transceiver	Transceiver	Transceiver
Package	QFN24	QFN40	Proprietary
Size (mm)	5 x 5	6 x 6	20 x 20
Supply (V)	1.9 - 3.6	2.0 - 3.6	1.9 - 3.6
Max. Power (dBm)	0	4	4
Sensitivity (dBm)	-80	-86	-85
Max. Current (mA)	22	34.3	17
RAM (KB)	NA	NA	NA
ROM (KB)	NA	NA	NA
Sleep Current (μA)	2	0.5	0.5
Price Estimate (€)	5.06	2.54	11.96

NA - *Not Applicable*
UA - *Unavailable*

The ANT chipsets encapsulate the full complexity of the wireless protocol, allowing resource constrained 4-bit or 8-bit MCUs to manage the formation and maintenance of large wireless networks using the ANT technology. This encapsulation is only possible given the compactness of the ANT protocol. When compared to the technologies formerly presented (Bluetooth and IEEE 802.15.4), the availability of ANT transceivers is rather limited due to its emerging nature. However, as documented in Table 2.4, three manufacturers are currently offering transceivers that, together with an MCU, can be used to developed ANT+ wireless networking solutions. Nordic [79] supplies the nRF24AP1 transceiver, whose current consumption varies from 2 microamperes in the sleep mode to 22 milliamperes in the reception mode (peak). Texas Instruments [68] offers the CC257x line of transceivers, characterized by current consumptions ranging from 500 nanoamperes

in power down mode to 34.3 milliamperes when performing transmissions with 4 dBm of power. The ANTAP281M4IB transceiver module is a ready to use solution provided by Dynastream Innovations Inc. [80]. This module encompasses an ANT transceiver and all the external components (including antenna) required to establish communications with other enabled ANT devices. This module operates with currents from 500 nanoamperes in deep sleep mode to 17 milliamperes when actively listening for transmissions. As Table 2.4 illustrates, the cost of the ANT transceivers is characterized by a wide variation. Also, only the Texas Instruments CC257x family of transceivers presents a competitive cost when compared to the former Bluetooth and IEEE 802.15.4 alternatives.

nanoNET

nanoNET [81] is a proprietary low-power wireless communication technology developed by Nanotron Technologies GmbH [82]. It targets battery-powered mobile communication and real-time location applications in the fields of factory automation, intelligent access control and alert systems, among others. This technology operates in the ISM 2.4GHz band and is advertised as being highly immune to noise due to the employed Multi Dimensional Multiple Access (MDMA) modulation method together with the Chirp Spread Spectrum (CSS) technique. The MDMA combines three modulation schemes (frequency, amplitude, and phase) into a single method that wastes no bandwidth and is able to provide a given level of Quality of Service (QoS). The CSS spread spectrum technique uses robust *chirp* pulses that, in addition to reducing the required transmission power for a given transmission distance, is highly immune to multipath fading. The CSS technique was originally developed by Nanotron Technologies GmbH and integrated on the IEEE 802.15.4 standard in 2007 [75]. The nanoNET MAC sublayer defines access schemes such as ALOHA, Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA), Time Division Multiple Access (TDMA) and CSMA/CA over TDMA, thus potentiating a wide range of applications.

To the best of our knowledge, Nanotron is the only manufacturer developing chips that make use of the CSS spread spectrum technique. On this account, Table 2.5 presents the main characteristics of the nanoPAN 5375 transceiver module manufactured by Nanotron. This device combines the function of location awareness with wireless communication in a single chip. Additionally, the device encompasses not only the transceiver chip, but also all the electronic components required to establish wireless communications driven by an external MCU. It supports three power and transmission modes: HC Ranging, LD Ranging and R Comm. The first offers high-capacity ranging with communication and provides a

80 MHz bandwidth together with a 1 Mbps data rate. The second, refers to long-distance ranging with communication and supports 80 MHz of bandwidth and a data rate of 250 kbps. The third is tailored to allow robust communications only. This mode of operation is characterized by a bandwidth of 22 MHz and a data rate of 250 kbps.

Table 2.5: nanoNET commercial solutions

Parameter	nanoPAN 5375
Type	Transceiver
Package	Proprietary
Size (mm)	15 x 29
Supply (V)	2.3 - 2.7
Max. Power (dBm)	20
Sensitivity (dBm)	-96
Max. Current (mA)	210
RAM (KB)	NA
ROM (KB)	NA
Sleep Current (μA)	UA
Price Estimate (€)	93.9

NA - *Not Applicable*
UA - *Unavailable*

As documented in Table 2.5, the nanoPAN 5375 transceiver module has a current consumption of 210 milliamperes when performing transmissions at its maximum power of 20 dBm. Although the transmission range is significantly increased with the support for 20 dBm transmissions, its application in battery powered applications with demanding autonomy requirements seems difficult, given that the current consumption can be of one order of magnitude higher than competing communication technologies. Furthermore, although the nanoPAN 5375 module adds ranging for supporting RTLS applications together with wireless communications, it presents a much higher price tag when compared to its communication competitors (e.g., Bluetooth and IEEE 802.15.4). The RTLS specialization and the lack of massification of this technology are the main reasons contributing to its increased cost.

Following a description of the key enabling communication technologies operating in the 2.4 GHz ISM band, a review of the main wireless real-time protocols devised for this band is provided in the following subsection.

2.2 A Review on Selected Wireless Real-Time Protocols

The MAC layer has been acknowledged to play a critical role in providing timing and reliability guarantees in real-time protocols [83]. In this sense, the analysis of such protocols should focus on their MAC features and operation. Kumar *et al.* [84] classify protocols according to their MAC schemes in contention-free and contention-based protocols. In contention-free MAC schemes, bandwidth is usually reserved for each transmission, thus avoiding contention among nodes to access the medium. This type of MAC protocols is typically employed in scenarios with centralized control, where a node (or more) establish the rules of access to the medium that the remaining must comply with. Contention-based MAC schemes are characterized by a risk of occurring collisions among data transmissions. This type of MAC protocols is more geared towards dynamic networks, where the control to access the medium is distributed among nodes. In the following, a review on selected wireless protocols is presented. The review encompasses protocols using both types of MAC schemes, although with a much higher predominance of contention-free protocols. Due to their demanding timeliness and reliability requirements, the selected protocols focus on factory automation applications. Also, only communication protocols operating in the 2.4 GHz ISM band were considered, given the requirement of global support. In addition, the selected protocols target applications with a static nature regarding its deployment and operation. The following description provides, for each selected protocol, a brief introduction followed by an analysis of its operation with focus on the adopted MAC scheme. A discussion of the protocol's timeliness performance and limitations is also presented.

2.2.1 WISA

The Wireless Interface for Sensors and Actuators (WISA) protocol [85, 86] was proposed under the assumption that existing off-the-shelf wireless systems could not cope with the power supply and timeliness requirements of communications at the machine level in factory automation. In this sense, the WISA technology provides both wireless communication support and wireless power supply to automation devices. Since batteries are not an option in industrial applications encompassing hundreds of devices, the power supply is obtained using a magnetic coupling technique where energy is transmitted using long-wave radio frequencies. The real-time communication is implemented with a combined scheme of TDMA and frequency hopping on top of the physical layer of the IEEE 802.15.1 protocol.

The WISA technology encompasses two types of devices: sensor/actuator node and

base station. The former employs a hardware platform that benefits from the "economies of scale (low cost), component integration (small size) and low-power consumption" [85] of using a standard communication technology, in this case a Bluetooth 2.4 GHz radio transceiver. Thus, it employs a commercial IEEE 802.15.1 transceiver, a receiver coil for the wireless power supply and a microcontroller. The later adopts a specialized architecture encompassing an RF front-end, microcontroller and an FPGA, which is responsible for the time critical control of the baseband signal processing. The requirement of using a dedicated FPGA for signal processing is related to the base station requirement of operating in full-duplex mode, besides receiving signals from up to four sensor/actuator nodes simultaneously.

Provided that WISA protocol uses the same physical layer of Bluetooth, it shares its physical characteristics, as described in Section 2.1.1. For example, the nominal transmission power is of 0 dBm and the raw data rate is of 1 Mbps.

Operation

The WISA protocol is characterized by a medium access mechanism that combines techniques such as TDMA, FDD and FH to ensure communications with fast response times, high reliability and ability to serve a large number of devices deployed in small areas. As documented in Figure 2.8, time is divided in long intervals called WISA frames, each one lasting 2048 microseconds. These frames can be of two types: downlink and uplink. The former, as the name suggests, is used by the base station (BS) to convey information to sensor/actuator (SA) devices. This frame is segmented into 16 downlink slots, each one used to perform the transmission of a packet with a length of 128 bits. The structure of this packet is specified in Figure 2.8. Among other fields, it contains the data for 8 SAs. Downlink slot transmissions are continuously performed within each WISA frame in order to keep the synchronization of the SAs with the BS timing.

Uplink slots are used by SA devices to conduct their uplink transmissions. A WISA time frame is segmented in 32 slots, with a duration of 64 microseconds each. However, because the packets transmitted by the SAs on these slots only have a length of 56 bits, the remaining slack of 8 microseconds is used to establish a guard interval for transients. Uplink slot transmissions are performed by the SAs only when there is (data/control) information to be transmitted. Furthermore, these transmissions are only performed in the uplink slots specifically reserved for them, at a maximum rate of one SA transmission per WISA frame. Two guard slots are reserved at the WISA frame boundary to provide enough slack time

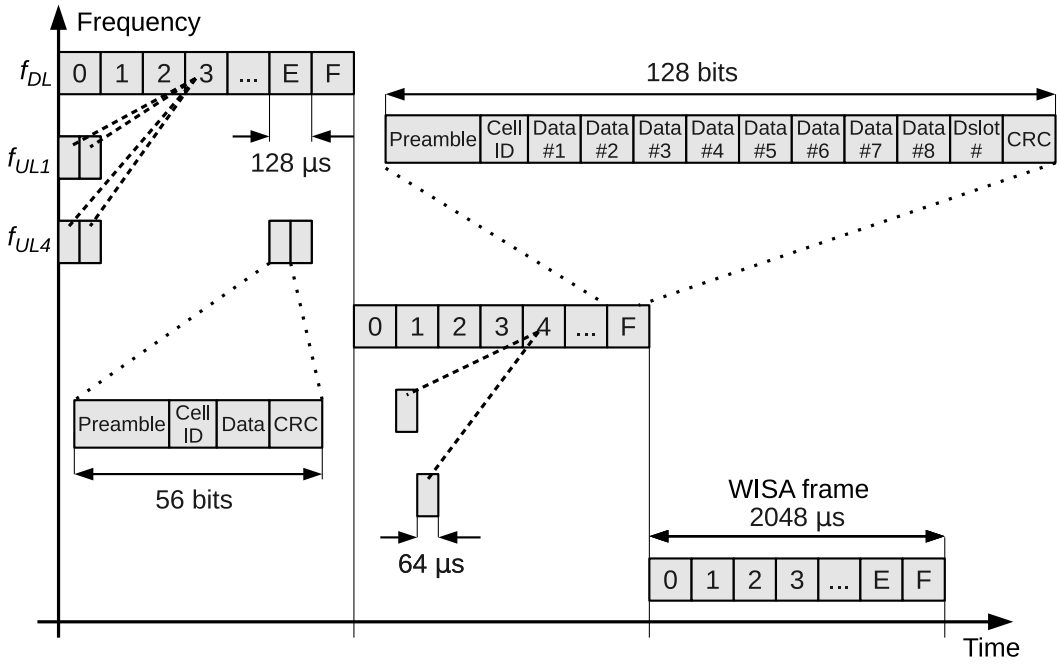


Figure 2.8: WISA TDMA/FDD/FH pattern

for the frequency hopping switching procedure. Hence, only 30 uplink slots per uplink group are available. Provided the use of four uplink groups (UL1, UL2, UL3, UL4) in the WISA protocol, it is possible to support up to 120 sensor/actuator devices per cell.

As depicted in Figure 2.8, the downlink and the uplink groups use a different frequencies (f_{DL}, f_{UL1}, f_{UL4}), which hop synchronously in each frame boundary. The available ISM 2.4 GHz band is segmented in 7 disjoint sub-bands, each one containing 11 hop frequencies spaced 1 MHz apart. Thus, an individual hop frequency can be chosen by arbitrating a 7-ary integer that identifies the sub-band and a 11-ary integer that selects the individual frequency within the chosen sub-band. The frequency hopping is conducted so that consecutive frames are transmitted in different sub-bands spaced by, at least, 11 MHz; uplinks belong to the same sub-band, but are spaced 3 MHz apart from each other; and the minimum separation between downlink and uplink frequencies is of 22 MHz.

An automatic retransmission request (ARQ) scheme is used together with Frequency Hopping (FH) in order to improve the WISA communication performance. In this sense, after a transmission in the reserved uplink slot, the SA waits for an acknowledgement in the corresponding downlink slot. If the acknowledgement is not received, the packet is retransmitted on the reserved slot of the next frame. Otherwise, the SA receives the acknowledgement and verifies that the packet was successfully received by the BS.

Performance and Limitations

The WISA protocol performance was evaluated using a realistic testbed consisting of 60 WISA SA devices placed around a WISA base station, at a distance of around 70 centimeters [85]. The SA devices were configured to generate a data packet at a rate of two packets per second. Results are provided solely in the form of a normalized number of transmission attempts. Hence, given that the TDMA framing is bounded by a maximum of 2048 microseconds and that any lost packet (uplink/downlink) results in an additional delay of 2048 microseconds (due to its retransmission in the subsequent WISA frame), it is possible to estimate the maximum delay experienced by any SA transmission. The performance analysis of the WISA protocol focused on three scenarios encompassing different sources of noise, namely: co-located WISA cells; industrial environment equipments; and other communication systems.

In the first scenario, a multi-cell operation was simulated by pointing the antennas of a variable number of interfering BSs (1 to 3) to the WISA system testbed. Results demonstrated that the maximum number of retransmission attempts is 4 and occurs in the scenario encompassing 3 interfering BSs. This value is well within the design goal of 6 retransmissions, which can result in a maximum delay of 15 milliseconds.

The performance analysis associated to noise from industrial environment equipments was evaluated in both spot and arc welding installations, given their ability to generate very high electromagnetic field strengths, which can interfere with electronic equipment. The WISA sensors were placed near the welding guns, just a few centimeters away. Provided that the majority of the noise generated by welding equipment fades out above 1 GHz, results showed no measurable impact of this type of noise on the communication performance.

Regarding the performance of the WISA protocol when other communication technologies generate noise in the 2.4 GHz band, two technologies were evaluated: Wi-Fi and Bluetooth. In the first case, a PC with an IEEE 802.11g card was connected to an Access Point (AP) on channel 5, employing a 20 dBm transmission power. Measurements were conducted by placing the AP at three different distances away from the WISA BS (7.5, 4 and 1 meter). Results revealed a general increase in the number of retransmissions, with a maximum of 5 retransmissions occurring when the AP is placed 1 meter apart from the WISA BS. On the opposite direction, the WISA testbed had an impact on the WLAN performance. In the noise-free scenario, the WLAN data rate was measured to be of 3.1 MBps. However, by decreasing the distance between the WISA BS and the AP, the WLAN

data rates were also reduced to 2.8 MBps (7.5 meters), 2.25 MBps (4 meters), and 0.86 MBps (1 meter).

In the Bluetooth noise scenario, a laptop was configured to continuously transmit packets to a Bluetooth AP using a transmission power of 0 dBm. The Bluetooth interferer was placed 1 meter away from the WISA testbed. Results indicate a maximum of 3 retransmissions when the Bluetooth interferer is at a distance of 1 meter. Furthermore, the impact of the WISA testbed on the Bluetooth link was also measured. Results exhibit a reduction of the Bluetooth data rate from 29.2 kBps (noise-free scenario) to 25.6 kBps when the link is exposed to the WISA testbed.

The WISA protocol heavily relies on a specifically developed based station that plays a central role in the protocol's operation. This fact poses two relevant issues. The first is the requirement to develop custom hardware supporting the intended features, thus making difficult its replacement and significantly increasing the cost of the solution. However, because a single base station can support up to 120 wireless sensors/actuators, the cost becomes less significant in large wireless networks. The second issue reports to the existence of a single point of failure, which hinders a graceful degradation of the network operation in case of a malfunction.

2.2.2 TDMA-Based MAC Protocol for Industrial WSNs

Phua *et al.* [87] proposed a TDMA-based MAC protocol for Wireless Sensor Networks (WSNs) using *Link State Dependent Scheduling* (LSDS) to cope with the channel fading phenomena experienced in industrial environments. This protocol works by collecting samples of the channel quality and generating prediction sets in independent slots. The prediction sets are then used by stations to wake up and transmit/receive during the clear predicted slots while keeping in sleep mode during the slots predicted as potentially resulting in signal fading. The authors claim that this protocol improves the packet throughput when compared to both non-link state dependent TDMA and CSMA protocols.

Operation

In order to improve the reliability and efficiency of data transmissions in industrial WSNs Phua *et al.* [87] devised a TDMA-based MAC protocol incorporating a LSDS approach for each independent slot. This protocol was developed under the premises that, in industrial plants, signal fading occurs in approximately periodic time intervals and that

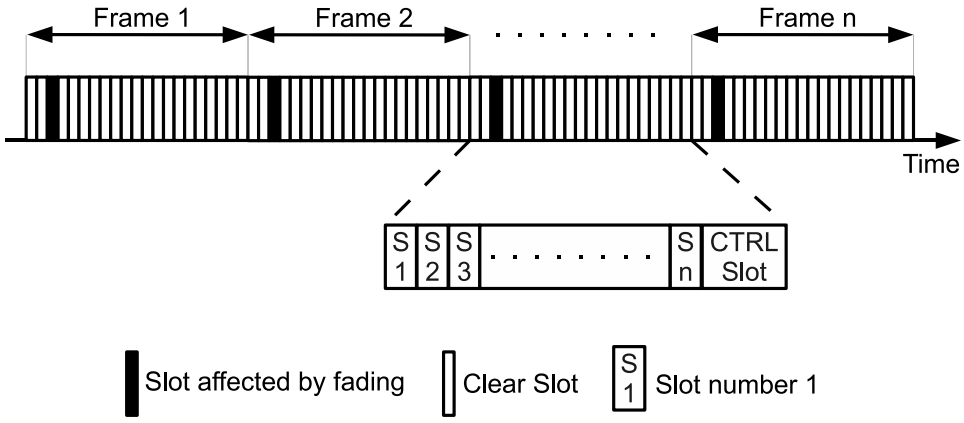


Figure 2.9: TDMA frame format

stations placed near each other are unlikely to be affected by it. Thus, signal fading affecting these stations is mainly caused by the periodic movements of nearby machines.

Figure 2.9 depicts the channel time divided in TDMA frames of fixed length ($1...n$), each one encompassing a set of communication slots ($S1...S_n$) and a control slot. Communication slots, as the name implies, are used for communication between sensor stations and are used in a pairwise fashion, i.e., each slot allows only one *uplink* (transmission) and one *downlink* (reception) between a pair of sensor stations. The control slot at the end of each TDMA frame enables the future development of the protocol to cope with the inclusion of new stations.

The protocol assumes a set of connected static sensor stations plus a base station. All sensor stations are connected to exactly one parent and each station can have one or more children. The base station, however, has no parent as it is the root of the network. Additional sensor stations can be added to the network after the deployment phase and stabilization. Initially, in a so called *startup phase*, stations operate in random access mode and exchange tokens and TDMA slot assignments. In this phase, the protocol employs a "bottom-up tree-based token passing approach" in which, because the whole structure of the network is known, the base station and the relay stations reserve the slots required to support exclusive communication for their children and descendants. Afterwards, stations exchange TDMA schedules with their neighbors and initiate the communications in TDMA mode.

The switching from random access mode to TDMA mode is immediately followed by an evaluation period called *observation phase* where a training data set is collected for a given period of time and later used to compute a prediction set allowing error-free

transmission/reception of data packets. In the observation phase, sensor stations analyze the fading signal pattern in order to obtain its length and period. This is achieved by performing transmissions during the scheduled slots negotiated in the startup phase. As the sensor stations can detect when a scheduled transmission was not received, they can register the occurrence of *good* and *bad* slots related to the successful or unsuccessful reception of the scheduled packets in a given slot. Later, this information is used by the receiving station to build a *prediction set* which is the subset of the training set with the shortest repeat pattern. This prediction set is then transmitted at the beginning of the *signal fading aware* (SF-Aware) phase in an uplink slot to the source station (for that slot). In this phase, the transmission and reception of data in a given slot is only conducted if the prediction set indicates a clear slot. Further information regarding this protocol can be found in [88].

Performance and Limitations

The protocol relies on the premise that fading occurs in strictly periodic intervals. However, in practical perspective, this is not likely to happen. A continuous deviation is expected to occur over time, perhaps with different rates along the day. This deviation would severely impair the performance of the protocol once the fading period becomes significantly dephased from the TDMA frames/slots. Although a simple mechanism of forcing an observation mode systematically could be devised, this would be very costly as the protocol is complex and could take a non-negligible amount of time to conclude this task.

Although the authors present supporting simulation results regarding the protocol's effectiveness in predicting channel fading, there are several open issues that should be addressed concerning a physical implementation, namely: the interference impact that other technologies may have in the devised mechanism's reliability; the foreseeable timeliness with and without external interference; the technology selection for its implementation; and, finally, the protocol's ability to support operational flexibility.

2.2.3 Real-Time Sensor/Actuator Network for Factory Automation

In 2007, a real-time Wireless Sensor/Actuator Network (WSAN) protocol targeting factory automation applications and operating in the 2.4 GHz ISM band was proposed by

Körber *et al.* [89]. This protocol was designed with focus on eight user driven requirements, namely: autonomous energy operation; 5 milliseconds end-to-end bounded delay; multisensor/actuator handling of at least 64 nodes; fieldbus comparable reliability; coexistence with existing wireless standards; scalability and modularity; low-cost hardware architecture supported on COTS components and modules; and global market support. The authors performed an indoor radio channel characterization concluding that the protocol's underlying physical layer should support a data rate of 1 Mbps.

The architecture of the proposed network is based on network cells, due to its scalability and direct application in factory environments where production processes are segmented into independent subprocesses. The network itself is characterized by a star topology, having a base station (BS) that bridges the WSAN to the (cabled) upper tier of the network. The BS encompasses a transceiver for establishing cabled communications in the upper tier and an array of wireless transceivers, included to support the simultaneous transmission of packets in multiple frequencies. Sensor/actuator modules (SAMs) typically employ a single wireless transceiver, but can use two, if full duplex communications between BS and SAMs are required.

Operation

In this protocol, the access to the medium is managed using a classical TDMA approach, where the transmission time is divided into slots that are characterized by a bounded interval and a specific transmission frequency assigned individually to each SAM. This mechanism avoids collisions among transmissions from nodes belonging to the network. Provided that the BS encompasses multiple transceivers, it is possible to support varying levels of throughput, robustness and timeliness in each factory cell.

Figure 2.10 depicts the timing diagram of the protocol. As documented, the superframe is initiated with the BS transmitting a beacon packet that synchronizes the SAMs and defines the length of the network's communication cycle, corresponding to 6 milliseconds. Among other information, this packet encapsulates configuration data targeting the SAMs and is provided by the network's control application. Since each SAM is assigned with an individual communication slot, it may perform its transmission on the assigned slot after receiving the beacon packet without risking a collision with another SAM. However, provided the stringent autonomy requirements, it may remain in sleep mode for long periods of time (not transmitting in the designated slot), waking up only in predefined intervals to transmit an "alive" packet.

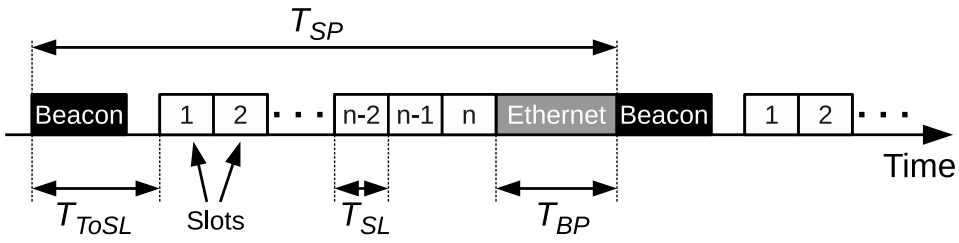


Figure 2.10: WSAN protocol time diagram

The interval between the beginning of the beacon and the beginning of the fist SAM slot is designated "time to slot" (T_{ToSL}). It allows the SAM performing the transmission in the first slot to have enough time to process the beacon packet. The "beacon preparation time" (T_{BP}) is enforced to allow the communication with the control application through the Ethernet connection.

Performance and Limitations

The performance of this real-time WSAN protocol was assessed using a testbed encompassing an upper level control unit (ULCU) together with a base station and 18 sensor/actuator modules. The ULCU was built using a C coded control application running in a PC. As introduced, the BS is responsible for managing the communications with 18 SAMs by means of periodically transmitting the beacon packet. The ULCU's PC is connected to the BS by means of an Ethernet connection that is used to transport information among these devices encapsulated in UDP packets compliant to the PROFINET protocol.

Regarding the timeliness performance of the WSAN protocol, results demonstrate that under (optimal) laboratory conditions the end-to-end delay varies from 6.1 to 11.1 milliseconds for remotely accessing the 18 SAMs. The presented testbed is capable of enabling the occurrence of simultaneous transmissions in two different frequencies, which allows improving the network's timeliness and throughput. However, under harsh environmental and electromagnetic interference conditions, it is foreseeable that these results will significantly deteriorate. Hence, in order to plan the deployment of this technology in factory settings, a coexistence assessment with the most widely adopted technologies in 2.4 GHz ISM band (IEEE 802.11, IEEE 802.15.4 and Bluetooth) is of paramount importance.

This real-time sensor/actuator network can be analyzed according to three other aspects. First, due to its centralized nature, the base station is a single point of failure. This means that a malfunction in this device causes the whole network to halt, thus impair-

ing any system based upon it. Second, because the protocol was not designed towards providing operational flexibility, it lacks the support for on-line adaptation mechanisms, thus increasing its exploitation and maintenance costs. Finally, being a custom designed protocol, it lacks the benefits of using standard protocols such as the wide availability of devices supporting it, for example.

2.2.4 WiDom

The Dominance Protocol for Wireless Medium Access (WiDom) [90] was developed to adapt the dominance or binary countdown protocol used in technologies such as the Controlled Area Network to a wireless channel. Dominance protocols assign to each station of the network (or message) a given unique priority. Before initiating a transmission, a station waits for a specific amount of time, before the channel is idle, and then starts an arbitration phase where each station contending for the channel sends its unique priority in a bit-by-bit fashion, starting with the most significant one. Building on the fact that the transmission of a dominant bit overwrites the transmission of a recessive bit, a station that is transmitting a recessive bit but detects an ongoing dominant bit transmission will back off from the arbitration procedure as it detects the contention of a higher priority station. This mechanism guarantees that if all stations have different priorities then, at the end of an arbitration phase, only one will win the contention and proceed with the desired data transmission.

Although inspired by the dominance or binary countdown mechanism, the WiDom protocol faced nontrivial problems regarding its wireless nature. First, the implementation of the dominance protocol in a wired medium is achieved using a wired AND approach, something that is not possible when using a wireless medium. Second, the physical implementation of the dominance mechanism relies on the possibility of simultaneously performing transmissions and listening to the medium state to check whether the transmitted bit is overwritten by another station, which is usually not possible using wireless technologies.

Provided that when a station is transmitting a dominant bit there is no need to sense the medium and, given that when a recessive bit is to be sent, no bit is required to be effectively sent, only the medium is to be sensed, the WiDom protocol uses these characteristics to transmit the priority in the arbitration phase of the protocol named *tournament*. During this phase, the transmission duration of a bit takes a fixed amount of time, which is much longer than the one of a data bit. Nevertheless, a station winning the contention can transmit the data packet at the maximum speed.

Operation

The WiDom protocol operates in three phases: *synchronization*, *tournament* and *transmit/receive*. Figure 2.11 illustrates a WiDom contention cycle between two stations S1 and S2. The synchronization phase occurs from instants t_1 to t_4 and t_1 to t_5 for station S1 and S2, respectively, while the tournament phase elapses between instants t_4 and t_6 for S1, and between instants t_5 and t_7 for S2. The transmit phase of station S1 (contention winner) occupies the period of time starting at instant t_6 and ending at instant t_8, while the reception phase of station S2 (contention looser) occurs between instants t_7 and t_8.

The synchronization phase aims at establishing a common reference time for all stations wishing to transmit. During this phase, stations wait for a (long) F period of time to ensure that no station disrupts an ongoing tournament. Afterwards, two scenarios can occur: there is a pending message to be transmitted or not. The first scenario occurs for station S1 having an enqueued message at t_2. As such, after a period of time denoted by E, a carrier pulse signaling the beginning of the tournament phase is transmitted with a duration given by $H + SWX$, where H represents the duration of the carrier pulse and SWX the time required to switch between transmission/reception mode. In the second scenario, the station will perform a carrier sense for an H period of time. Meanwhile, a message may have been dequeued for transmission or not. In the later case, given that no message is waiting to be transmitted, the station will not participate in the following tournament phase and it will be set to carrier sense mode. In the former case, as occurs for station S2, a message has been dequeued during the H period at instant t_3 and the contention for the medium must be performed in the following tournament phase.

The tournament phase is characterized by the transmission of the priority, having a specific number of bits, in this case three (3). As reported, since both stations S1 and S2 have messages to be transmitted, they both content for the medium. However, between the transmission of carrier signals, stations must wait for a *guarding* period of time (G) so that clock inaccuracies can be tolerated. As such, before initiating the transmission of the priority's most significant bit, stations S1 and S2 wait for this amount of time. Following, assuming that station S1 has a priority 5 (0b101) and that station S2 has a priority 4 (0b100), both stations initiate the transmission of their priorities keeping a guarding time G between carrier signals. If a station contends with a dominant bit '1' (carrier transmission) then it will win the contention for that bit. Conversely, if the station contends with a recessive bit (carrier sensing) and detects a carrier wave then it looses the contention and proceeds to listening to the medium to acknowledge which priority has won

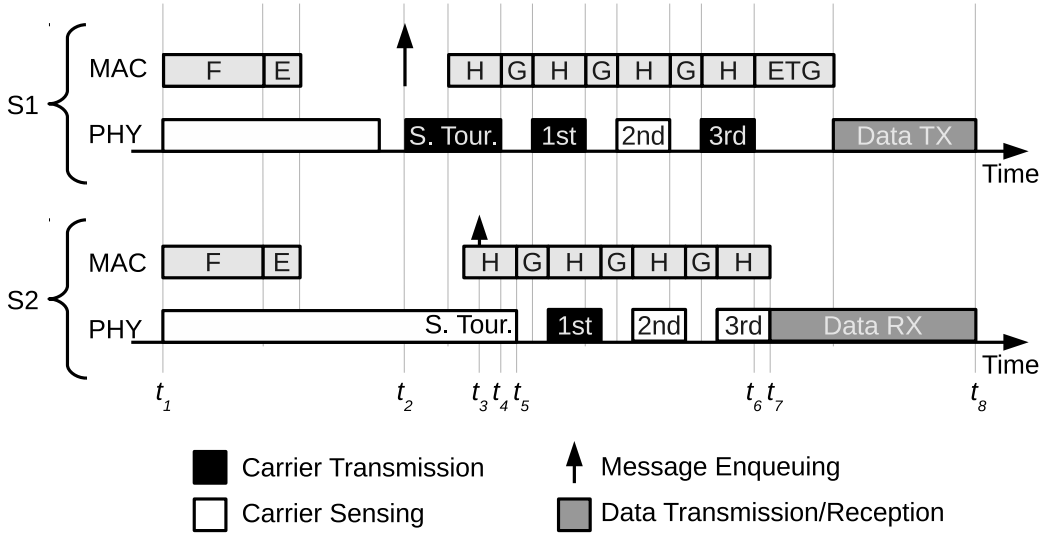


Figure 2.11: WiDom MAC and PHY protocol activity

the tournament.

The end of the tournament phase results in a single winning station, in this case station S1, given that it has the higher priority ($5 > 4$). Following, the winning station has to initiate the transmission of the data message. However, it can only proceed with this task after waiting for a fixed period of time ETG, allowing all stations to switch to receive mode, as depicted in Figure 2.11. Further information regarding the WiDom protocol and the different constants F, E, H, SWX, G and ETG is available in [90].

Performance and Limitations

The WiDom protocol allows a prioritized medium access within bounded time, which is the basis for supporting wireless real-time communications. The protocol's authors developed a test setup that enables assessing its performance, both in terms of robustness and timeliness. Regarding the former, it was observed that the probability of correct reception and prioritization of messages is 100 % when using 2 or 10 nodes placed in a circumference with a 1 meter diameter. However, when the diameter is increased to 4 meters, this probability decreases to 99.998 %. In this sense, it is expected that, with an increased diameter, e.g., near the typical nominal range of the IEEE 802.15.4 technology (10 meters), the probability of a correct reception and prioritization of messages is even more reduced.

The timeliness assessment of the WiDom protocol encompassed two different traffic sce-

narios: periodic and sporadic. In both scenarios, trials with a different number of messages transmission requests are performed (ranging from 1 to 10) and the associated response time is registered. Table 2.6 summarizes the measured WiDom response time bounds presented in [90]. The "Best Case" corresponds to the trials performed with only one message stream, while the "Worst Case" occurs in the trials having 10 message streams contending for medium access. As documented, the delay is always higher than 25 milliseconds and the jitter is several times larger than the message response delay, even without any relevant interference from other contending technologies. Hence, the support of real-time applications with demanding timeliness requirements regarding latency and jitter such as, for example, industrial process control seems far beyond the capabilities of this protocol. In fact, because "if a node experiences strong noise (...) then the protocol will simply lose its tournament and not start any new ones during the duration of noise" [90], there is a high risk of node starvation due to intentional or unintentional interference from co-located networks operating in the same region of the spectrum.

Table 2.6: WiDom response time

	Sporadic		Periodic	
	Delay (μs)	Jitter (μs)	Delay (μs)	Jitter (μs)
Best Case	25752	71458	25752	92986
Worst Case	27627	208159	27627	151909

Another aspect that should be considered is the fact that the WiDom protocol relies on the transmission of specific carrier signal during the tournament phase, which can only be implemented using very specific hardware. In this sense, provided the specificity of the signal, it may result in a manufacturer lock-in, thus increasing costs and limiting the implementation.

2.2.5 RT-Link

The RT-Link protocol [91, 92] aims at providing real-time wireless communications in industrial control, surveillance and inventory tracking applications. It employs specific out-of-band synchronization hardware to enable a collision free medium access, which results in an improved utilization of the medium, timeliness and energy efficiency. The authors of the protocol identify the support of synchronized and collision-free communications

as the two fundamental challenges in providing bounded delay services for applications requiring energy-efficiency. These challenges are met through the design of a TDMA-based link layer protocol supported on two different synchronization hardware solutions: an Amplitude Modulation (AM) carrier-current transmitter/receiver for indoor applications and an atomic clock receiver for outdoor scenarios. In the first case, an AM transmitter periodically broadcasts a pulse, which is received by stations encompassing an add-on low-power AM receiver module. This pulse allows to globally synchronize the clock of the receiving stations. In the second case, each station includes a WWVB atomic clock receiver module allowing global synchronization in outdoor environments. The authors claim that, besides providing predictable network lifetime together with bounded end-to-end delay, the RT-Link protocol offers an increased flexibility when compared to random access protocols, namely in what concerns the control of the topology, the independence of the delay regarding the sampling rate and the on-demand multi-rate support.

Operation

The RT-Link protocol was designed to support two types of stations: fixed and mobile. The fixed type relates to the infrastructure support and, as such, encompasses out-of-band synchronization hardware to ensure the medium access timing coherence. The mobile type, on the other hand, relates to devices whose localization varies over time and, thus, will synchronize by listening transmissions from the fixed stations and does not include specific synchronization hardware. Although behaving differently regarding time synchronization, both fixed and mobile stations share a common base of hardware encompassing microcontroller, IEEE 802.15.4 transceiver and sensors.

In indoor scenarios the RT-Link protocol synchronization is conducted using a carrier-current AM transmitter, which takes advantage of the power wiring of the building to radiate a time synchronization pulse to all stations in the neighborhood. This transmitter is fed with an atomic clock pulse in order to maintain synchronization with any stations placed outdoor. The AM module attached to the fixed stations receives the periodic pulse and drives an input pin of the microcontroller, which triggers a time update.

Figure 2.12 represents the synchronization pulse (from t_1 to t_2) initiating a fixed length communication period denoted by *Time-Sync Cycle* ranging from t_1 to t_5. This pulse is followed by two frames, one containing *Scheduled Slots* (t_2 to t_3) and the other one containing *Contention Slots* (t_3 to t_4). The length of each slot corresponds to the time required to transmit a maximum sized packet. Scheduled slots, as the name suggests, are

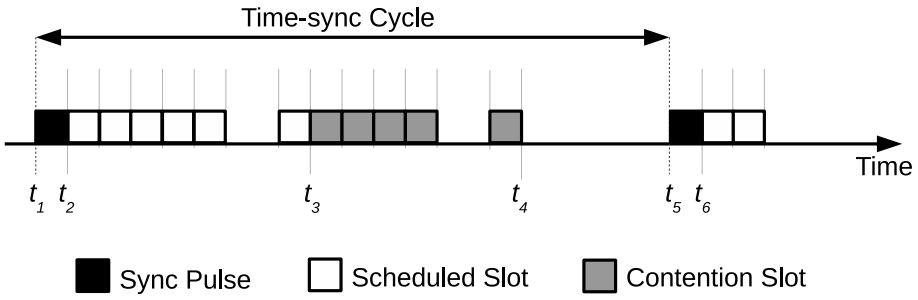


Figure 2.12: RT-Link time slot allocation

reserved for scheduled transmissions from given stations while contention slots are targeted for stations contending for the channel using a slotted-Aloha mechanism. On a timeliness perspective, nodes performing transmissions in the scheduled slots have a collision free medium access and, therefore, enjoy a guaranteed access to the medium. Conversely, the nodes that contend for the medium are prone to interference from other nodes wishing to transmit in the same slot, which makes it difficult to provide timeliness guarantees.

As illustrated in Figure 2.12, after the last contention slot, between instants t_4 and t_5, the station schedules itself to wake up just before the following synchronization pulse and switches to a low-power consumption mode. Afterwards, this cycle repeats. Further information regarding the RT-Link protocol and, specifically, about the network formation and packet transmission scheduling can be found in [91].

Performance and Limitations

The key result reported by the authors and supported by the presented measurements is the ability of the RT-Link protocol to offer a predictable network lifetime on par with a bounded (multi-hop) end-to-end delay. This is possible due to the use of a global time synchronization mechanism that is described to be both "economical and convenient for indoor and outdoor deployments" [91]. However, the adoption of a global synchronization approach using out-of-band synchronization hardware implies additional hardware, which in turn, increases the cost of the solution as well as the required power. Besides, the adoption of a WWVB atomic clock signal for outdoor synchronization limits its deployment to the United States of America, given the general lack of support for this synchronization signal worldwide.

Provided that the RT-Link protocol is based on the IEEE 802.15.4 technology for data communications, it can be exposed to intentional or unintentional interference in the 2.4

GHz ISM band. In this sense, in order to evaluate the feasibility of deploying networking solutions for factory automation settings based on this protocol, an experimental evaluation of its coexistence is critical. Such coexistence analysis should assess the RT-Link protocol performance under packet transmissions (interference) generated by co-located IEEE 802.11, IEEE 802.15.1 and IEEE 802.15.4 technologies on overlapping channels, given their broad market penetration.

2.2.6 Wireless Fieldbus for Plastic Machineries

A wireless proprietary protocol for plastic machineries was proposed by Flammini *et al.* in [93]. At the core this protocol is the requirement of supporting up to 16 wireless devices with a maximum cycle time of 128 milliseconds. The protocol's design goals were focused on minimizing the overhead; reducing the amount of information exchanged; lowering the amount of energy spent; increasing the protocol's efficiency; and decreasing the computational effort. Hence, the authors based their solution on an IEEE802.15.4 compliant transceiver for implementing the physical layer of the protocol together with an hybrid Medium Access Control (MAC) layer supported on both Time Division Multiple Access (TDMA) and Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). The former, in addition to providing a smaller communication overhead, allows to guarantee the required cycle time deadline of data transmissions. The latter is suited for exchanging network management packets, given its aperiodic nature.

Regarding the topology, the protocol adopts a star topology, provided that the factory setting addressed by the authors is characterized by a set of nodes deployed relatively close to each other. Hence, there is no need for complex routing, which permits a simplification of the network layer implementation. In this star topology, nodes are only allowed to communicate with a special device named network coordinator. Provided that nodes only employ one transceiver, they can only perform packet transmissions in a channel at a time. Therefore, multiple subnets can be deployed together and exploit frequency diversity. The instantiation of the proposed protocol was conducted for a plastic machining application, encompassing a set of wireless thermocouples, which include a simple IEEE 802.15.4 compliant transceiver paired to a 8-bit microcontroller that is capable of performing the analog data acquisition.

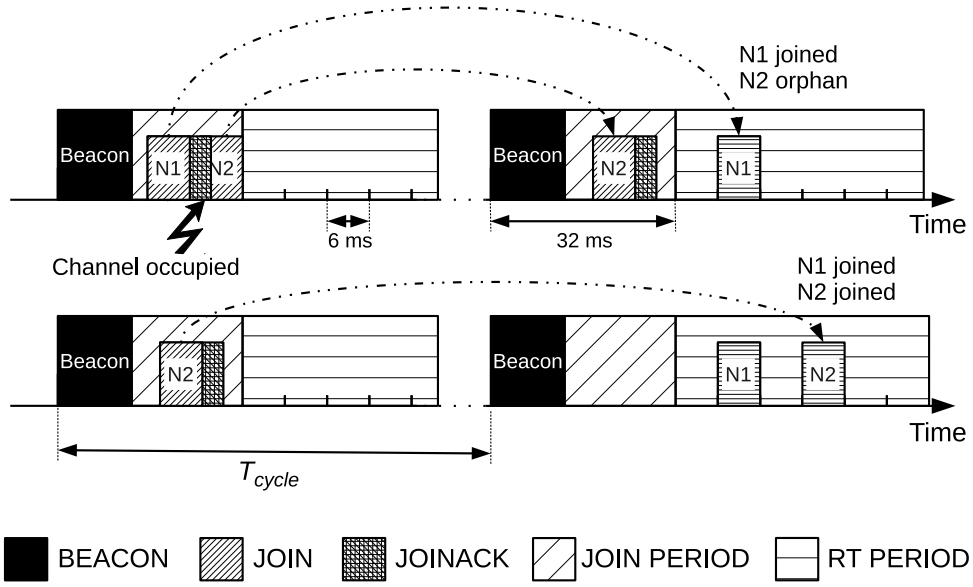


Figure 2.13: CSMA/CA and TDMA hybrid medium access

Operation

The network coordinator lies at the center of the proposed network because it is responsible for periodically transmitting the BEACON packet, which triggers the beginning of a new communication cycle. Due to the need of remaining active for long periods of time, it is mains powered. As depicted in Figure 2.13, the interval of time following the BEACON packet is reserved for network formation (JOIN PERIOD), allowing nodes to join by transmitting JOIN messages.

The network coordinator assigns communication slots in the real-time period (RT PERIOD) by means of JOINACK messages. The duration of the network formation window (32 milliseconds) was chosen to accommodate a sufficient number of packets for this purpose in each communication cycle. Furthermore, the window enabling real-time communications by using a TDMA medium access approach has a duration of 96 milliseconds. This window defines 16 slots with 6 milliseconds of length each. A wireless device participating in the network with a slot attributed must remain active during the duration of that slot. However, only a fraction of this time is effectively used to perform the actual transmission ($\approx$ 1 millisecond), after the settlement of the analog circuits pertaining to the thermocouple interface (1 millisecond). The remaining time provides a slack timing period named interslot time interval that lasts approximately 4 milliseconds.

The protocol employs a delayed acknowledgement strategy envisaging both the energy

consumption and the medium occupancy reduction. In this scope, the BEACON packet encompasses a bit array field containing the individual acknowledgements of the packets sent in the preceding communication cycle. Moreover, the authors exclude retransmissions based on the knowledge that the "physical quantity of interest is oversampled due to a small minimum cycle time" [93]. This approach allows some samples to be lost without affecting the performance of the system. Also, because the noise in wireless channels is bursty by nature, the approach of dropping retransmissions is more suitable, since closer retransmissions would have a higher probability of failing.

Regarding channel diversity, building on the coordinator ability to measure the radio frequency activity on different channels, the protocol adopts a strategy of classifying the channels into two groups according to their floor noise and selecting the best channel of each group to become the communication channel and the backup channel, respectively. In both cases, the BEACON packet defines, which channel shall be used in the communication cycle. When a device fails to listen the BEACON packet for a period longer that 8 cycles, it starts listening the communication and the backup channels alternatively for an interval of 8 communication cycles. If this period elapses without receiving a BEACON packet, the nodes initiates a BINDING procedure, looking for an active coordinator.

Furthermore, the authors describe a comparable solution based on the IEEE 802.15.4 standard and employing the Guaranteed Time Slots (GTSs) provided in the beacon enabled mode. These slots can be reserved by the network coordinator for devices requiring a specific bandwidth, in this case, wireless thermocouple nodes. Without going into much detail, the necessary modifications are addressed. First, since the IEEE 802.15.4 standard allows a maximum of seven GTS per superframe, an alternating (round-robin) GTS assignation policy is required to interleave two node groups, each one encompassing up to seven sensors. Using this approach, it is possible to provide a similar throughput to the proposed solution. Second, the Beacon Payload field of the Beacon frame is used to convey the acknowledgement bitmap required to implement the proposed delayed acknowledgement.

Performance and Limitations

Regarding the assessment of the protocol implementation timeliness, two approaches were addressed. In the first, the timing correctness of the protocol was successfully evaluated using a custom sniffer, capable of generating a digital signal marking the packet transmissions on a given IEEE 802.15.4 channel. In the second, the protocol timeliness and reliability were analyzed. A testbed encompassing four wireless thermocouples was

installed on a plastic injection moulding machine for this purpose. Provided that in each cycle (128 milliseconds) a new temperature value was generated by the wireless thermocouple, the protocol supported approximately 8 samples per second for each of the four nodes. The authors claim that the packet's associated latency is "virtually null".

The network reliability was assessed using the progressive sequence numbers embedded in the transmitted packets to check for gaps. Five trials with a duration of 1 hour each were conducted in different days, with the plastic injection molding machine operating normally. Results demonstrated no sensor rebinding procedure nor switching from the communication channel to the backup channel. However, some packets were lost during the trials, namely: a maximum of four consecutive BEACON packets at a sensor; a maximum of seven consecutive DATA packets (from the same sensor) at the coordinator; and a maximum of there consecutive DATA packets (from the same sensor) at the sniffer.

Given the centralized nature of this protocol, upon a failure of the network coordinator, the global wireless fieldbus halts its operation due to lack of beacon packets. Hence, if the system being controlled has no redundancy, this failure may result in a significant production loss. Furthermore, provided that the protocol operates in the popular 2.4 GHz ISM band, it is susceptible to being exposed to interference from contending wireless technologies. In this sense, it is likely that its reliability and timeliness performance will be affected by high levels of noise in the communication channel.

2.2.7 ISA SP100.11a

The ISA SP100.11a standard [94] was originally designed to provide reliable and secure wireless communications supporting non-critical monitoring and control applications in industrial process settings [95]. It was approved in 2009 by the International Society of Automation (ISA) and became the IEC 62734 standard [96] in January 2014 after being voted and approved by the Technical Committees 17, 22 and 57 of the International Electrotechnical Commission.

The ISA SP100.11a defines several types of devices, namely: field devices, routers, backbone routers, and gateways. Field devices are input/output (I/O) devices, which have the minimum characteristics that allow them to participate in an ISA SP100.11a network. These devices typically operate so as to maximize energy conservation. Routers, as the name implies, must have routing capability, be able to act as proxy and support clock propagation. These devices are responsible for increasing the communication's robustness, allowing the extension of the network, and supporting differentiated QoS for multiple traffic

flows. The backbone router is a special type of router, whose operation is focused on the network's backbone and on the support for the encapsulation of external network protocols. The gateway typically placed at the edge of the ISA SP100.11a network and provides the necessary interface between this and other networks. As in the WIA-PA protocol (see Section 2.2.9), the ISA SP100.11a protocol defines two logical devices: system manager and security manager. The system manager is responsible for the network's configuration, performance and operational status, while the security manager handles all the aspects of the network pertaining to security.

The ISA SP100.11a standard defines a five layer OSI compliant protocol stack. At the top of the stack is an optional object-oriented application layer, which can be used to enable object to object communications. This layer defines the interoperability and interaction mechanisms that support object communication and provides the means for establishing protocol tunneling, thus enabling the access of legacy protocols to the devices of a ISA SP100.11a network. The key features provided by the transport layer pertains to the support of end-to-end communications based on connectionless services. The network layer is mainly responsible for the routing mechanism at the backbone level and for the fragmentation/reassembly of packets going to/coming from the data link layer. The ISA SP100.11a medium access control and the subnet level routing are defined in the data link layer. It encompasses three sublayers supporting a) a subset of the IEEE 802.15.4 MAC features; b) an extension of the IEEE 802.15.4 standard with new CSMA/CA, TDMA and frequency hopping mechanisms; and c) routing at the subnet level. The ISA SP100.11a physical layer is based on the IEEE 802.15.4 standard, thus operating in the 2.4 GHz ISM band.

Several network topologies are supported by the ISA SP100.11a, namely: star, hub-and-spoke, mesh or a combination of them. The star topology is characterized by presenting the lowest possible latency across the physical layer due to to its single hop architecture. The hub-and-spoke topology is built with field devices around backbone routers, which are then connected to the gateway. Finally, the mesh topology is supported on routing devices, which extend the network coverage by supporting connectivity with multiple hops. In this topology, path diversity can be employed to improve the network's reliability.

Operation

The ISA SP100.11a protocol defines that devices are organized in subnets and that transmissions are performed in specific time intervals (slots) according to a specific sched-

ule provided by the system manager. The time slots are synchronized among devices belonging to the same subnet, i.e., devices in a given subnet have their slots aligned. At the application level, devices wishing to perform transmissions must request network resources to the system manager, supplying information about their communication requirements. Using the provided requirements (data and timeliness) information together with the information about the performance and connectivity between node pairs, the system manager calculates graphs and specific time slots to sender/receiver pairs of devices, thus enabling their communication.

The access to the medium is based on a TDMA scheme in which time is segmented in slots of configurable duration (typically 10/12 milliseconds). The use of 12-millisecond timeslots is related to the support of the duocast mechanism, which requires an additional interval of 1-2 millisecond for an additional acknowledgment. The duocast mechanism is intended to improve the network's reliability. This slot duration also provides a time extension that allows the prioritization of messages and the communication between devices whose slot synchronization is delayed by a maximum of 2 milliseconds. Timeslots can be either dedicated or shared. The former are reserved for contention-free communication, while the later are qualified for contention-based communications using a backoff algorithm similar to the one employed in CSMA/CA. Transactions in each timeslot are ruled by templates, which define the timing for the operations pertaining to the transmission/reception of packets (e.g., reception wait time, turnaround time, etc.).

In order to improve the coexistence of the ISA SP100.11a protocol with other technologies operating in the 2.4 GHz ISM band, channel-hopping is employed in the ISA SP100.11a protocol (see Figure 2.14). This technique is coupled with mechanisms of channel blacklisting and whitelisting, which prevent the channel-hopping sequence from visiting frequencies being used by other technologies and enables the reuse of previously blacklisted frequencies that have become available.

Besides presenting the delimitations of superframes (cyclic collections of equal length timeslots), Figure 2.14 depicts the three types of channel hopping defined by the ISA SP100.11a protocol: slotted, slow and hybrid. In the first, transactions per slot are performed on a different channel according to five pre-programmed default hopping patterns. Each slot uses a different channel and encompasses the transmission of a packet and its acknowledgement. The slow hopping groups a set of contiguous timeslots on a single radio channel. Hence, the mode still allows channel hopping, but with a longer switching period, ranging from 100 to 400 milliseconds per hop. The hybrid channel hopping adopts both

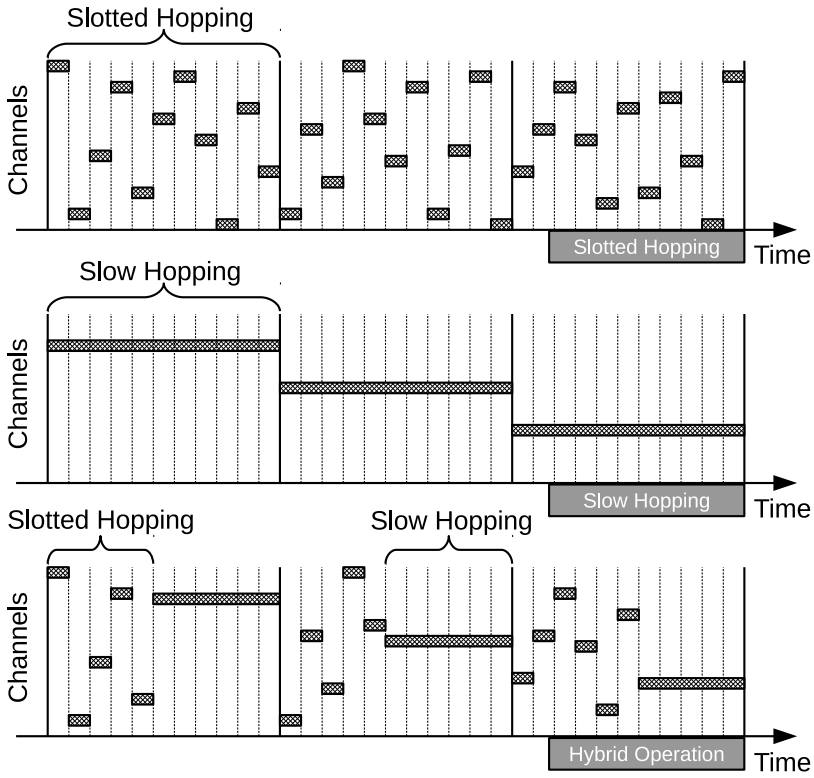


Figure 2.14: ISA100.11a frequency hopping

slotted and slow-hopping periods in a configurable combination.

Performance and Limitations

In [97], Wagner and Barton presented a comparative study between the MAC level performance of the ZigBee (CSMA-CA, single channel) and ISA100.11a (scheduled, channel-hopping) technologies under Wi-Fi traffic on a highly reflective, enclosed environment. The study focused on application-level message delivery rates as the primary indicator of performance. In the following, provided that the focus of this subsection is simply on the ISA100.11a protocol, only the relevant information pertaining to this technology is presented.

The setup used to assess the ISA100.11a performance under Wi-Fi noise was characterized by encompassing five wireless nodes, one gateway, one laptop computer (Wi-Fi traffic generator) and one Wi-Fi router configured to operate exclusively according to the IEEE 802.11g standard. The setup was installed within a steel cylinder experimental environment having approximately 3 meters of diameter and 6 meters of length. The ISA100.11a hops

through all the available 16 channels and the IEEE 802.11g interference was generated on channel 6 with several throughputs, namely: no interference, 5 Mbps, 10 Mbps, 20 Mbps. The wireless nodes were configured to perform transmissions at every 1, 5, and 10 seconds during an interval of one hour. The trials were performed three times and the percentage of successful packet deliveries was presented. The results of this study showed a consistent delivery rate above 99 % in all interference and data period scenarios, which, in the authors' opinion, demonstrates the frequency-hopping ability of the ISA100.11a technology to find free channels.

Shin and Rezha performed a simulation analysis of a hybrid network encompassing both CAN and ISA100.11a nodes in [95]. The study was realized using the OPNET modeler to simulate a network encompassing 6 CAN nodes and 10 ISA100.11a nodes connected to a gateway in a star topology. The gateway, as the name suggests bridges both networks. Nodes from both networks were configured to operate with a data rate of 250 kbps and the ISA100.11a superframe length aggregated 25 slots of 10 milliseconds each. The ISA100.11a nodes can be configured with dedicated slots from 1 to 24. The slots not reserved for deterministic traffic were allocated as contention slots. The detailed configuration of the CAN traffic is omitted as it is outside the scope of this analysis. The performance of the CAN-ISA100.11a hybrid network was evaluated with emphasis on its delay characteristics upon being subject to different network loads ranging from 1 to 9 packets per second from each node, following a poisson distribution. The results provided by the authors indicate that the path between the ISA100.11a node to the gateway contributes in excess of 95 % to the total experienced end-to-end delay. This fact was justified by the use of shared ISA100.11a slots, which employ the timeslot duration as a backoff period to resolve collisions.

The results mentioned above indicate that the ISA SP100.11a frequency-hopping algorithm is capable of successfully finding free channels, thus enabling a high level of reliability. However, at the time of this writing, no information about the performance of the the ISA SP100.11a technology in highly polluted scenarios is available.

2.2.8 WirelessHART

WirelessHART [98] was the first open wireless communication standard designed specifically to target process measurement and control applications [99]. The initial release of the HART protocol occurred in the 80s with a physical layer supported on a 4-20 milliamperes analog signal loop, which allowed reliable bidirectional communication with field instru-

ments. The evolution since then has been dramatic, not only due to the added features (security, diagnostics, etc.), but also because of the wireless capability added in the HART field communications protocol version 7. This protocol was named WirelessHART and it is compatible with existing HART devices and applications.

In general terms, WirelessHART is a secure and robust mesh networking technology built upon the IEEE 802.15.4 2.4 GHz ISM physical layer. It employs a channel hopping mechanism applied on a packet by packet basis. This channel hopping mechanism allows establishing a higher frequency diversity, thus contributing to increase the WirelessHART reliability. Furthermore, a blacklisting mechanism is also adopted to avoid channels being used by contending technologies. The WirelessHART protocol adopts a Time Division Multiple Access (TDMA) scheme to arbitrate and coordinate communications among the devices of the network. Hence, it follows a time synchronized approach where devices only establish communications in pre-scheduled timeslots, enabling granting QoS to transmissions, a critical aspect in process measurement and control applications.

The WirelessHART protocol encompasses a centralized network manager that is responsible for scheduling the transmissions of the network. This procedure is accomplished by considering the global network routing information together with the communication requirements provided by the devices and the application. The schedule defines which devices communicate in which timeslots. When the schedule is computed, the devices participating in the network receive only the information pertaining to them, i.e., they are only provided with the slots for which they have transmit or receive requirements. The network manager continuously monitors the network topology and communication demands, accordingly adapting the network graph and schedule.

Operation

The WirelessHART protocol employs a MAC compliant with the IEEE 802.15.4 (2006) standard [74], which enables its implementation with readily available Commercial Off-The-Shelf (COTS) IEEE 802.15.4 transceivers. As introduced, the original IEEE 802.15.4 MAC functionality is extended with TDMA and frequency hopping schemes, enabling the provision of collision-free and deterministic communications. At the base of such schemes is the concept of time slot, which is an interval of time with a duration of 10 milliseconds. The WirelessHART protocol also defines the concept of superframe as a sequence of contiguous time slots. Each superframe is characterized by the number of slots that it includes, which also defines its period. Figure 2.15 depicts three superframe cycles including 4 slots (TS0 to

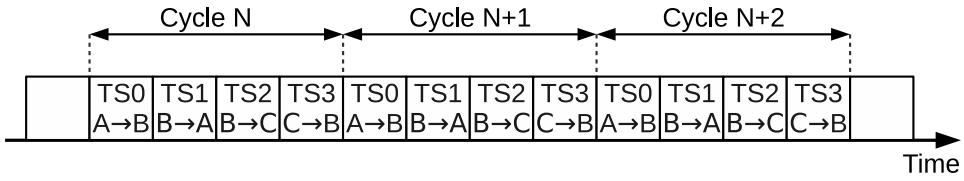


Figure 2.15: WirelessHART superframe example

.

TS3) each. The duration of each slot only allows one transmission and the corresponding acknowledgement to occur. As documented, the first two slots (TS0 and TS1) are used to perform communications between Device A and Device B in both directions. The third and fourth slots (TS2 and TS3) are used in the same way but by devices B and C. As shown, the superframe schedule is repeated in every cycle. Additionally, since a superframe is defined by a given set of channels and time slots, multiple superframes of different sizes may coexist in a given WirelessHART network. In this sense, different schedules with varying duty cycles can be assigned to different groups/types of devices. For example, it is possible to allocate superframes encompassing multiple QoS requirements to different device groups.

The operation of a mesh network employing a TDMA scheme relies heavily on a precise time synchronization, since all communications must occur within the designated time slots. In this sense, all devices participating in the network must share a common notion of time, i.e., they must perceive the beginning and ending of slots with minimum variations throughout the network. As introduced, the duration of a timeslot (10 milliseconds) was selected to allow sending/receiving one packet in a given channel and the associated acknowledgement. Furthermore, as documented in Figure 2.16, the timeslot also includes periods of time to allow the synchronization of the WirelessHART network.

Figure 2.16 shows that the receiver must start listening the medium after *TsRxOffset* time units after the beginning of a time slot [100]. Furthermore, the receiver must acknowledge a received packet within *TsMaxPacket* + *TsTxAckDelay* time units after the beginning of the packet. In order to maintain the time synchronization among the WirelessHART network devices, time is propagated periodically outward from the gateway. The underlying synchronization mechanism is as follows. Individual devices are constantly adjusting their network time using the Time Adjustment field carried by the acknowledgement packets received from its time source neighbors. However, if packets are not exchanged for long periods of time with these neighbors, the synchronization fails. As a consequence, each

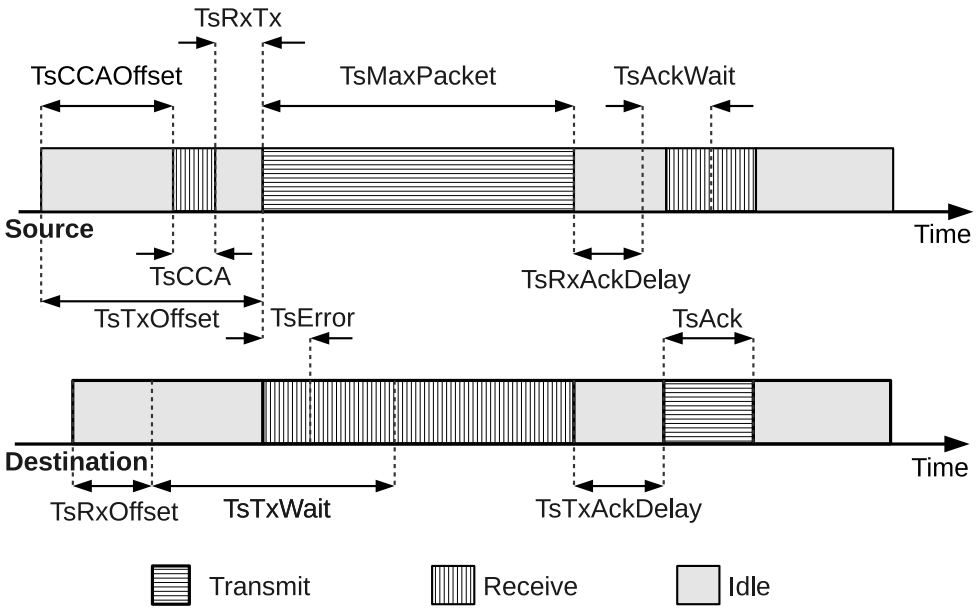


Figure 2.16: WirelessHART slot timing

device encompasses a keep-alive timer programmed with the maximum inactive interval that still guarantees the synchronization with the network's time. When this timer expires for a given neighbor, a keep-alive packet to this neighbor is automatically generated, thus forcing a time resynchronization.

Performance and Limitations

In [101], besides studying the time synchronization of a WirelessHART network, Ferrari *et al.* evaluated the end-to-end latency of the HART command 1 "Read Primary variable" using a setup with commercial devices. The time synchronization was assessed using a testbed that allows either one or two hops between the gateway and the WirelessHART device under test. In both scenarios, the period deviation was observed for an interval of 50 time slots. Results demonstrated an increased jitter in the two-hop scenario together with 50 % of the slots having a synchronization error centered around 0.1 milliseconds, which is the maximum allowed offset between clocks. The end-to-end latency was assessed using a setup that allows measuring both the time elapsing between the "Read Primary variable" HART request and the corresponding response (using Wireshark); and the interval between the over-the-air WirelessHART request and the acknowledgement reception of the WirelessHART response packet (using a WirelessHART probe). The corresponding values were, on average, 3 seconds and 1 second, respectively.

Petersen and Carlsen studied the performance of a WirelessHART network in an industrial environment in [102]. The performance was evaluated, among other parameters, in terms of latency. The setup used to perform the assessment consisted on a network of 9 sensors and 1 gateway. The WirelessHART network testbed was configured to operate with a superframe encompassing 150 timeslots, resulting in a superframe with a duration of 1500 milliseconds. Three scenarios were evaluated: interference "clear"; interference from IEEE 802.11g networks; and interference from a chirp jammer. The interference "clear" scenario was evaluated over a period of 120 hours. Network latency results showed an average value of 2037 milliseconds during the entire test period with a standard deviation of 92 milliseconds. This value arises from the length of the superframe and the latency introduced by the retransmissions of lost packets in the multi-hop mesh network.

The scenario considering interference from IEEE 802.11g networks was assessed over a period of 24 hours. This scenario was further segmented in two sub-scenarios with different WLAN beacon intervals: 100 milliseconds or 20 milliseconds. In the first case, the average latency observed was of 1885 milliseconds with a standard deviation of 99 milliseconds. In the second case, the average latency observed was of 2760 milliseconds with a standard deviation of 315 milliseconds. These results indicate that "deploying one or more WLAN access points in the same area as a WirelessHART network will lead to a degradation of the WirelessHART network" [102].

Finally, the scenario where the WirelessHART network is exposed to interference from a chirp jammer was tested over a period of one hour. The jammer sequence consisted on the transmission of a time-varying noise signal with a sweep period of 10 microseconds. The signal was activated after 15 minutes of the beginning of the test period and was active for 45 minutes. Results showed that in the initial phase of the jamming sequence the network is completely halted with no packets being received and, thus, with no valid latency. These results corroborated the notion that DSSS based networks are generally "vulnerable to time-varying spectral noise, as both the spread signal and the jammer interference have large bandwidths" [102].

From the results presented above, it is possible to conclude that the timeliness of the WirelessHART protocol is highly dependent on the superframe duration and slot configuration, which enables a limited tailoring of its operation to different QoS requirements. Hence, it is constrained regarding its operational flexibility, which difficults the adaptation of its behavior to different application requirements. Furthermore, although it employs a blacklisting mechanism allowing a higher level of reliability, it displays a significant per-

formance degradation in the presence of interference. Hence, a dramatic reduction of its performance is foreseeable in open environments characterized by a strong level of contention in all available channels.

2.2.9 WIA-PA

The Wireless Networks for Industrial Automation - Process Automation (WIA-PA) [103] is the Chinese standard for industrial wireless communications for process automation. After WirelessHART (see Section 2.2.8), WIA-PA was the second industrial wireless communication standard in the world (IEC 62601) specifically addressing industrial wireless communications. As the name implies, this standard addresses production processes, particularly, its activity monitoring and control support.

The WIA-PA standard adopts a hybrid star-mesh network topology. The bottom level of the network encompasses the so-called field and handheld devices. Field devices, as the name implies, are installed in the factory floor in order to monitor/control sensors/actuators while handheld devices are used in the management of the network. The bottom level is built around a star topology (cluster), where some nodes (routers) act as cluster heads. The routers are responsible for forwarding packets between WIA-PA network devices. The upper level is realized by means of a mesh topology that integrates both routers and gateways, which are devices that bridge the WIA-PA network and other plant networks with the purpose of translating the protocols and mapping data. In addition, the WIA-PA standard defines two logical devices: network manager and security manager. While the first is usually implemented at the gateway and is responsible for the configuration, communication scheduling and performance monitoring of the network, the second is responsible for establishing and maintaining the WIA-PA security mechanisms.

The WIA-PA communication protocol defines a full stack of layers, including Application Layer (APL), network layer (NWK), Data Link Layer (DLL) and Physical Layer (PHY). The APL includes objects for aiding the implementation of industrial monitoring and control applications, besides providing a mechanism to manage the system and the available network resources. The NWK layer supports the network formation, route discovery and the static packet routing in multi-hop networks, among other features. The DLL handles the channel access and aspects such as the topology, point-to-point communications, etc. In this regard, although the channel access is compatible with the IEEE 802.15.4 standard, it was extended to cope with the requirements of industrial applications, as it will be seen further ahead. The WIA-PA PHY is defined by the IEEE 802.15.4

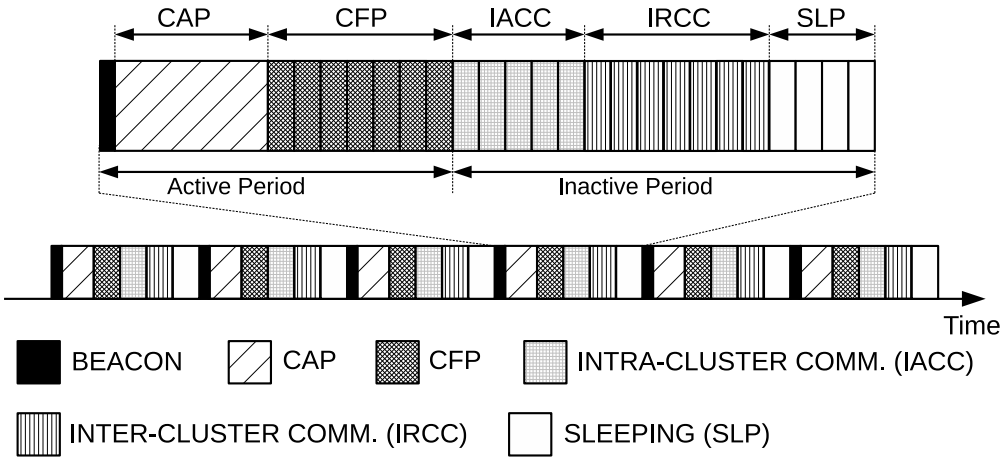


Figure 2.17: WIA-PA superframe

standard.

Operation

A superframe structure of the WIA-PA protocol is documented in Figure 2.17. As presented, it is based on the IEEE 802.15.4 standard beaconed mode frame structure presented in Figure 2.5. It encompasses an active and an inactive part. The former, includes a Contention Access Period (CAP), used to allow devices to join the network, perform intra-cluster management, among other activities. Furthermore, it also embeds a Contention Free Period (CFP) where deterministic communications between field devices and cluster heads can be scheduled. The inactive part of the WIA-PA superframe is used to extend the IEEE 802.15.4 protocol, allowing intra and inter-cluster communications. The inactive period can also be used to reduce power consumption by switching to sleep mode.

In order to support real-time communications in industrial applications, two key IEEE 802.15.4 MAC additions are proposed by the WIA-PA protocol: extra deterministic slots and different priorities for non-periodic communications [104]. The former addresses the limitation of the number of GTSs in the CFP. Provided that the key concern in this application scenario is not power conservation, part of the superframe's inactive period can be used to provide additional guaranteed slots, i.e., an additional period of time encompassing time slots similar to those of the CFP. These slots can be scheduled by the network manager when there is no available GTSs in the CFP to be attributed. In this case, the field device assigned with the slot outside the CFP must remain active during the inactive part

of the superframe, going to sleep mode only after its designated time slot. The remaining field devices will go into sleep mode during the inactive portion of the superframe.

The priority-based contention for non-periodic real-time communications is adopted from the IEEE 802.11 standard. In this sense, different Inter Frame Space IFS time intervals are respected before initiating a packet transmission: urgent IFS (UIFS), RIFS for real-time IFS (RIFS) and nor-real-time IFS (NIFS). When a device listens to the medium before initiating a non-periodic transmission, it first waits for the channel to become idle. When that occurs, it waits a specific amount of time (IFS) that depends on the time of data being transmitted (urgent, real-time or nor-real-time) before sampling the medium again. After the expiration of the IFS interval, the medium is checked again. The CSMA/CA mechanism is then conducted using a modified exponential random backoff algorithm, where the backoff interval of the different types of non-periodic transmissions has different exponential increments and ranges, thus enabling higher priority communications to access the wireless channel first.

The WIA-PA protocol avoids interference by means of frequency hopping. In this sense, it defines three hopping mechanisms: Adaptive Frequency Switch (AFS), Adaptive Frequency Hopping (AFH) and Timeslot Hopping (TH). The first defines that, in a given superframe cycle, the channel is kept unchanged for the Beacon packet, CAP and CFP. Then, depending of the channel conditions, the channel can be switched on a subsequent superframe cycle by the network manager. The AFH scheme defines that the time slots in the intra-cluster period change irregularly the communication channel depending on the actual channel condition. Finally, the TH hopping scheme states that, to avoid interference and fading, the channel used per timeslot in the inter-cluster period is changed regularly.

Performance and Limitations

The performance of the WIA-PA protocol was theoretically analyzed by Zhong *et al.* in [104]. Two types of traffic were considered in this analysis: predefined periodic communication and non-periodic communication. In predefined periodic communications the end-to-end delay was found to be deterministic, bounded and minimized in both scenarios of intra and inter-subnet communication. Provided that deterministic time slots are assigned to periodic communications in either the CFP or in the inactive window's intra-cluster communication period, the queuing delays at the NWK and MAC layers are small, thus justifying these results. The performance of non-periodic communications was concluded to depend heavily of its priority, which defines the length of the IFS being used. Urgent

communications are those with the higher probability of being served. However, because the delay affecting these communications depends of the number of messages contending for the medium, they are not bounded as periodic communications. Non-periodic real-time and nor-real-time messages also contend for the medium with the urgent messages. However, because they have lower priorities, they wait for the end of the transmission of all urgent messages before contending among themselves for the medium. In this sense, their end-to-end delays are larger and remain unbounded.

The WIA-PA protocol employs a flexible channel hopping mechanism, which renders possible avoiding channels under heavy use. Nevertheless, it can still be negatively affected when exposed to environments with a reduced number of available "free" channels. Due to its novelty, there are no studies addressing the performance of the WIA-PA protocol under interference on the 2.4 GHz ISM band.

2.3 The 2.4 GHz ISM Band Hubbub

The Oxford Advanced Learner's Dictionary [105] defines a *hubbub* as a "loud sound made by a lot of people talking at the same time" or "a situation in which there is a lot of noise, excitement and activity". Given the current heterogeneity of communication technologies operating in the 2.4 GHz ISM band and their trend of broad adoption fueled by its unlicensed nature, a continuously increasing *hubbub* is to be expected in this region of the spectrum. Since different technologies sharing a frequency band may have an impact on each other, a review of the related work on the coexistence between local and personal area networks operating in the 2.4 GHz ISM band is provided.

2.3.1 IEEE 802.15.4/IEEE 802.15.4 Coexistence

The coexistence among IEEE 802.15.4 based networks has not been widely studied as IEEE 802.15.4/ WiFi and IEEE 802.15.4/ Bluetooth technologies. This mainly occurs because the underlying technology (physical layer) is common and it should be simple to configure the networks in order to make them uncoupled from each other. However, given both the existence of uncontrolled IEEE 802.15.4 stations and the adoption of frequency-hopping techniques built on top of the IEEE 802.15.4 physical layer, interference is likely to occur. The literature addressing the performance of the IEEE 802.15.4 protocol is typically focused on characterizing specific aspects such as, for example, packet delivery ratio, hop

delay and ratio of collisions between hidden terminals [106], or even end-to-end capacity [107].

This subsection focuses on some selected works addressing the impact of uncontrolled IEEE 802.15.4 stations transmitting in overlapping or adjacent channels of an IEEE 802.15.4 network. In [108], Bertocco *et al.* present an evaluation of the impact of ZigBee interference on beaconless IEEE 802.15.4 communications. The authors experimentally demonstrate that the interference caused by ZigBee transmissions has a higher impact than the transmissions performed by Bluetooth or IEEE 802.11g interferers. Results show a PER of 55 % when using IEEE 802.15.4 data stations configured to operate with CCA modes 1 and 2. This indicates that the transmissions performed by the IEEE 802.15.4 interferer reach the data stations with a power level above their threshold. The disabling of the CCA reduces the PER to 16.5 %, which is still a non-negligible value.

Lo Bello and Toscano analyzed the cross-channel interference occurring in co-located IEEE 802.15.4 industrial networks in [109]. The experimental evaluation was conducted using a beaconless ZigBee network and the corresponding analysis was focused on parameters such as PER, packet loss and delay. As expected, results from a single interferer employing a similar transmission power show that the effect of cross-channel interference is highly dependent of the interfering node channel. The spurious emissions of the interfering signal are mainly "felt" in near channels (with a minimum offset), where a maximum PER of 4.5 % may occur due to the IEEE 802.15.4 interference. However, if the power of the interfering IEEE 802.15.4 station significantly exceeds the one employed for valid transmissions, the PER escalates rapidly with an increase on the Signal-to-Interference Ratio (SIR), going above 60 % when the interferer is placed at 20 centimeters from the IEEE 802.15.4 network. When a valid IEEE 802.15.4 network is exposed to an increasing number of interferers on the same channel, the interference impact on that network decreases. This occurs due to the CSMA mechanism running on the interferers that increments the beacon exponent (reduces the generated traffic) upon the detection of a busy medium (by other interfering stations). The work presented in [109] also studies the effect of the minimum backoff exponent on delay under cross-channel interference. The authors conclude that it is "advisable to set the $macMinBe$ for high priority real-time traffic to zero" because it leads to shorter transmission delays with less jitter.

Nordin and Dressler investigated the effects and implications of beacon collisions in co-located IEEE 802.15.4 networks in [110]. The authors provide several simulations of two co-located IEEE 802.15.4 beacon-enabled networks, which are either in communication

range or in interference range (out-of-range) of each other. The performance analysis of these scenarios is focused on goodput, received beacons and number of collisions. Results indicate that the interference of beacons is a significant cause of performance degradation, although "the number of lost beacons is independent of the amount of superframe overlap" [110].

2.3.2 IEEE 802.15.4/Bluetooth Coexistence

As reviewed in Section 2.1.1, the Bluetooth technology follows a master-slave medium access scheme, employing a slow frequency hopping mechanism that ensures a permanence of a minimum of 625 microseconds in each channel of the hopping pattern. An adaptive frequency hopping (AFH) scheme is supported from the Bluetooth specification version 1.2 onwards. This mechanism allows improving the coexistence of the Bluetooth technology by excluding from the hopping sequence channels being used by other technologies. Nevertheless, until the AFH mechanism detects such channel activity and effectively blacklists those channels, Bluetooth may continue to use the full set of hopping channels and cause interference in co-located networks.

The impact of the Bluetooth technology on beaconless IEEE 802.15.4 based networks has been extensively studied in [111, 112, 113, 108, 114, 115, 116, 117, 118]. Conversely, the impact of Bluetooth transmissions on IEEE 802.15.4 based networks operating on the beacon-enabled mode has received much less attention in the literature. The most relevant work addressing this issue was published by Herrera *et al.* in [113]. Different versions of the Bluetooth technology were employed in the construction of the interferers. For instance, interfering devices compliant with the Bluetooth specification version 1.1 were only used in [113], while version 2.0 + EDR and version 2.1 + EDR Bluetooth devices where employed in [114, 115, 117] and [116], respectively.

In [112], Shin *et al.* model and simulate the Bluetooth interference impact on the IEEE 802.15.4 (ZigBee) PER. As expected, the authors conclude that "as the distance between ZigBee and Bluetooth increases, the PER of ZigBee decreases". Also, if the the distance between the ZigBee coordinator and the Bluetooth source is smaller than 80 centimeters, the PER ranges from 12 to 16 %. The PER together with the packet loss ratio have also been the performance parameters of choice for experimentally evaluating the IEEE 802.15.4 resilience to Bluetooth interference in several other research works [111, 108, 114, 115, 116, 117, 118]. Sikora and Groza [111] analyze the packet loss rate resulting from interference generated by co-located Bluetooth stations operating in the same spectrum region. Results

indicate a maximum packet loss rate of 10 % when an IEEE 802.15.4 link is exposed to Bluetooth interference. A study of the coexistence of IEEE 802.15.4 and Bluetooth for vehicle networks was performed by Zacharias *et al.* in [116]. Results demonstrate a maximum packet drop rate of 3.1 % and 7.45 % for Bluetooth traffic generated in SCO and ACL links, respectively.

Bertocco *et al.* [108] conclude that, provided its frequency hopping nature, Bluetooth interference does not disturb an IEEE 802.15.4 network, regardless of the CCA mode being employed. Francisco *et al.* [114] evaluate the coexistence of ZigBee networks and Bluetooth inside vehicles. Their investigation seems to corroborate the conclusions of Bertocco *et al.* [108], since the obtained PER is below 1 % for all nodes embedded in the vehicle. Also, the authors measure the average packet latency, which is always smaller than 3 milliseconds. A similar result is presented in [115], where the authors conclude the that ZigBee and Bluetooth networks "are able to work simultaneously with almost no impairment". Lavric *et al.* [117] and Guo *et al.* [118] present additional results reinforcing the conclusion that, for distances above 1 meter between a Bluetooth interferer and the IEEE 802.15.4 network, the interference impact is residual.

As introduced, the performance of the IEEE 802.15.4 beaconed and non-beaconed modes under Bluetooth interference is investigated by Herrera *et al.* in [113]. Several conclusions are drawn. First, the distance between Coordinator and End Device has a dominant impact on the PER over the existence/absence of Bluetooth interference, considering distances from 1 to 2 meters. Second, although the mean PER increases under Bluetooth noise, the beacon-enabled mode shows a higher susceptibility to this interference, since its mean PER value increases 130 % while, comparatively, the corresponding nonbeacon-enabled value only increases 66 %. Third, an increase of the distance between Coordinator and End Device above 3 meters has a negligible impact on the communication's PER under Bluetooth interference.

2.3.3 IEEE 802.15.4/Wi-Fi Coexistence

The interference caused by Wi-Fi transmissions on IEEE 802.15.4 networks has been one of the most investigated topics on coexistence among wireless technologies. The main motivation is the pervasiveness of the IEEE 802.11 technology among application domains where IEEE 802.15.4 based communication solutions can be deployed. Hence, provided that their coexistence characterization is of paramount importance, two main approaches were adopted. The first relies on effective implementations and experimental evaluations,

allowing a more realistic investigation of the coexistence issue. This approach, however, becomes onerous in terms of cost and labor when setting up a testbed for a complex evaluation scenario. Examples of this approach can be found in [111, 119, 120, 121, 122, 108, 123, 124, 125, 115, 116, 117, 118]. The second approach is established upon the development of models, which provide a representation of the real components in a communication system. Although generally providing less accurate estimates when compared to field tests, this approach is typically more flexible and easy to scale in terms of testbed setup. Several research works addressing the coexistence between the IEEE 802.15.4 and Wi-Fi technologies adopted this approach [126, 127, 120, 112].

Currently, commercially available Wi-Fi stations conform to one of the following specifications: IEEE 802.11b [128], IEEE 802.11g [129] and IEEE 802.11n [130], each one with specific characteristics. In the remaining of this section, a characterization of the key results and conclusions regarding the impact of these technologies on IEEE 802.15.4 communications is presented.

IEEE 802.11b interference

The interference caused by IEEE 802.11b stations on an IEEE 802.15.4 network has been analyzed by means of its impact on several parameters, namely throughput [126, 127], packet loss rate [111, 119], PER [127, 108, 112], PDF [108] and average delay [127].

In [111], Sikora and Groza address the coexistence between a beaconless IEEE802.15.4 network and a IEEE 802.11b network. Three key observations are described. First, as expected, the interference level caused by the IEEE 802.11b transmissions reduces with an increasing distance of the channels employed in both technologies. Second, the packet loss of the IEEE 802.15.4 technology exceeds 90 % when the transmissions of IEEE 802.11b stations are performed on overlapping channels. Third, the packet loss is not linear with the distance between Wi-Fi interferer and IEEE802.15.4 network.

On a similar line of work, Angrisani *et al.* [119] performed cross-layer measurements to characterize the coexistence between the IEEE 802.11b and IEEE 802.15.4 technologies. Their results show a degradation of the IEEE 802.15.4 packet loss ratio in excess of 50 % for Wi-Fi duty cycles above 40 %. Furthermore, this works also documents an increasingly higher packet loss ratio as a result of an increase in the power employed by the Wi-Fi transmissions. Finally, a key observation is presented by this study. It occurs when the Wi-Fi access point (AP) is distanced from the IEEE 802.15.4 network so that its transmissions are below the Wi-Fi's energy threshold. In this case, the "AP erroneously

assumes the channel free and transmits even when WSN is occupying the channel", thus causing collisions with the IEEE 802.15.4 packets.

Bertocco *et al.* [108] also addressed the coexistence between IEEE 802.15.4 and IEEE 802.11b networks, but with focus on the effect caused by the employed CCA modes. In this sense, the authors conclude that the use of the CCA1 mode is the most problematic concerning Wi-Fi noise. The justification is that the power perceived by the IEEE 802.15.4 nodes is frequently above the CCA threshold, thus hindering IEEE 802.15.4 transmissions from initiating, even if the SNR is far above the value required for a correct communication. This conclusion is supported by the presented PER and PDF results.

Chong *et al.* [126] provide an analytical study and a simulation assessment of the impact of IEEE 802.11b transmissions on the throughput of a ZigBee network. Several conclusions are drawn based on the obtained results. First, IEEE 802.11b transmissions can make the ZigBee communications practically impossible due to the severe throughput decrease for high loads of IEEE 802.11b interfering traffic. Second, for a given input load of IEEE 802.11b interference, the normalized throughput of a ZigBee network decreases as the length of the packets employed by the interferer increases. Third, the use of larger ZigBee packets makes the throughput more sensible to the IEEE 802.11b interference. Regarding the interference effect of IEEE 802.11b transmissions on the PER experienced by a ZigBee link, Shin *et al.* [112] present a simulation analysis whose results demonstrate that a PER increase is expected when there is an increment in the number of Wi-Fi stations or the distance between them and the ZigBee network is reduced.

In [127], Shin *et al.* provide an exhaustive analytical and simulation study addressing the interference occurring between IEEE 802.15.4 and IEEE 802.11b co-located technologies. A IEEE 802.15.4 network was configured to operate in the beacon-enabled mode using a slotted CSMA/CA medium access. This study corroborates the importance of the frequency separation between IEEE 802.11b and IEEE 802.15.4 technologies expressed in the obtained PER, latency and throughput. For example, it is demonstrated that for frequency separations smaller than 5 MHz, the PER is larger that 20 %, escalating above the 80 % threshold for frequency separations smaller than 3 MHz. Regarding latency, a 6.5 to 17.5 milliseconds increase is observed when the frequency separation between technologies is modified from 5 MHz to 2 MHz. In terms of throughput, a similar trend is documented. Shin *et al.* [127] also address scenarios where multiple IEEE 802.11b interfering stations perform transmissions. The results reported in these scenarios sustain two key ideas. First, the performance of the IEEE 802.15.4 network improves with an increasing distance to the

interfering IEEE 802.11b stations. Second, the performance of the IEEE 802.15.4 network worsens with an increasing number of interfering IEEE 802.11b stations. The author corroborates these conclusions with results for the PER, latency and throughput performance parameters.

IEEE 802.11g interference

The IEEE 802.11g [129] is, arguably, the most deployed WLAN technology today. In this sense, because it adds a new modulation scheme (OFDM) to the IEEE 802.11b version, it may interact differently with IEEE 802.15.4 based networks. Hence, its coexistence impact is important to be characterized. The evaluation of this technology was conducted under parameters such as PLR [122, 124, 125]; PDR [121]; RSSI [122, 118]; throughput and BER [120]; PER [117, 118]; LQI [118]; and FER [115]. The main conclusions regarding the impact of IEEE 802.11g transmissions on IEEE 802.15.4 networks are presented in the following paragraphs.

The performance assessment of IEEE 802.15.4 based networks under the influence of noise from IEEE 802.11g stations was widely studied using experimental approaches. Petrova *et al.* [121] conclude that the IEEE 802.15.4 channels overlapping the 20 MHz bandwidth of IEEE 802.11g channels experience a lower PDR as a consequence of a moderate/high traffic load of interference. For instance, an IEEE 802.11g interference traffic of 15 Mbps results in a PDR of around 60 %. Shuaib *et al.* [120] provide experimental results indicating a throughput drop of, a least, 10 % in an IEEE 802.15.4 link when exposed to IEEE 802.11g interference. These results also indicate that a smaller SIR results in a smaller throughput. For example, a 22 % drop in the throughput is to be expected when IEEE 802.15.4 stations are exposed to IEEE 802.11g interference while being separated by a distance of 12 meters (indoor).

Yang and Yu [124] also put forward results supporting the conclusion that the frequency offset and physical distance between networks is determinant in the PLR. As an illustration, the PLR drops from approximately 35 % to less than 5 % for frequency offsets of 2 and 8 MHz, respectively. In yet another example, the PLR drops from approximately 50 % to less than 12 % for distances of 1 and 8 meters between IEEE 802.11g AP and ZigBee coordinator. In [117] a similar trend is reported, but using the PER as the performance indicator. Bartolomeu and Fonseca [125] also experimentally demonstrated that, by performing IEEE 802.11g transmissions on a channel overlapping the frequency used by an IEEE 802.15.4 network, it is possible to severely compromise its reliability and, therefore,

impair all IEEE 802.15.4 communications. Musaloiu *et al.* [122] also analyze the impact of IEEE 802.11g interference on IEEE 802.15.4 sensor networks in both indoor and outdoor environments. The main conclusion is that an increase in the IEEE 802.11g data rate consistently results in both higher perceived RSSI values and higher IEEE 802.15.4 PLRs. Guo *et al.* [118] present an extensive analysis of the impact of IEEE 802.11g transmissions in the reliability of IEEE 802.15.4 WSNs deployed inside buildings. This work indicates that a maximum PER of 8 % can be reached depending upon the relative distances between transmitter, receiver and interferer station. Furthermore, the authors conclude that the computed LQI and RSSI values are not always coupled with the PER. These results are consistent with the conclusions presented by other authors.

IEEE 802.11n interference

The IEEE 802.11n [130] was the last wireless protocol to be widely adopted for WLANs. It builds on the existing family of IEEE 802.11 standards and adds two physical layer features, namely: support for Multiple-input Multiple-Output (MIMO) and 40 MHz channels. Provided the emergence and continuous adoption of this standard together with the inclusion of new physical layer features, the study of the IEEE 802.11n and IEEE 802.15.4 coexistence becomes highly relevant. The remaining of this section is dedicated to the analysis of relevant works covering this topic.

In [121], Petrova *et al.* perform an experimental analysis of the performance degradation between colocated IEEE 802.11n and IEEE 802.15.4 networks mainly focusing on PDR. The authors conclude that a moderate/high traffic load of IEEE 802.11n interference transmissions results in an IEEE 802.15.4 lower PDR when its channel overlaps an IEEE 802.11n channel, i.e., when the IEEE 802.15.4 channel central frequency lies within the bounds of the 40 MHz IEEE 802.11n channel's bandwidth. The results provided by the authors also support the conclusion that the residual side-band emissions negatively impact the PDR of co-located IEEE 802.15.4 networks. For example, considering an IEEE 802.15.4 CCA level of -77 dBm, a packet length of 26 bytes and a channel with a separation of -27 MHz to the IEEE 802.11n central frequency, the measured PDR is around 70 %.

A work featuring similar objectives and focusing on performance parameters such as the increase in the average loss rate and average latency was published later by Polepalli *et al.* [123]. In this study the coexistence between the IEEE 802.15.4 and IEEE 802.11n technologies was evaluated by configuring IEEE 802.15.4 transmissions in two different

IEEE 802.11n channels: extension and control. In the first scenario, results indicate a significant IEEE 802.15.4 performance loss for higher IEEE 802.11n traffic loads. For example, in the presence of a moderate IEEE 802.11n traffic load of 60 Mbps, the IEEE 802.15.4 average loss rate increases by approximately 30 % and the corresponding packet average latency increases by around 140 milliseconds. In the second scenario, a more dramatic effect is observed. The IEEE 802.15.4 average loss rate increases by around 80 % and the average packet latency increases by approximately 180 milliseconds.

A different coexistence perspective is presented by Zacharias *et al.* in [116], where an analysis is conducted considering vehicle networks. One key difference between this and other works on technology coexistence presented above is the use of a radio frequency anechoic chamber, which allows obtaining results without the influence of external interference or multipath propagation effects. The performance parameter used to evaluate the impact of IEEE 802.11n interference in IEEE 802.15.4 communications is the packet drop rate. The authors put forward results corroborating a maximum packet drop rate of 2.75 % when an IEEE 802.15.4 link is exposed to beacon traffic generated by an IEEE 802.11n station with a 102.4 milliseconds period.

2.4 Summary

This chapter presents the necessary background information for studying the feasibility of performing wireless real-time communications in open environments, where other technologies may contend for the medium. In this scope, several aspects are of particular relevance, namely: the existence of wireless technologies capable of enabling the support of real-time communications; the existence of protocols that are already capable of performing wireless communications and meet real-time requirements; and, finally, the potential limitations concerning their use in open environments.

In Section 2.1, the most widespread wireless low-power technologies operating in the 2.4 GHz ISM band are presented and discussed in detail, providing hints about their strengths and weaknesses. The Bluetooth and the IEEE 802.15.4 technologies have been selected as the primary candidates to support a wireless real-time communication solution capable of coping with the limitations of open environment deployment. Given the emergence of the ANT and nanoNET technologies, these have also been analyzed, but with less detail.

From a technical standpoint, when compared to Bluetooth, the IEEE 802.15.4 protocol provides a much higher degree of flexibility, both in terms of its network architecture and

timeliness support. The IEEE 802.15.4 protocol also provides a higher scalability, which significantly contributes to its use in a broader range of networking scenarios. The use of a DSSS spread spectrum technique together with the ability of employing frequency agility mechanisms to avoid interference makes the IEEE 802.15.4 standard an ideal solution to support reliable wireless communications in the 2.4 GHz ISM band. In an implementation perspective, the Microchip's MRF24J40 transceiver was selected mainly due to its low-cost (2.15 €), high sensitivity (-95 dBm) and relatively low current consumption (maximum current of 23 mA in active mode and 2 microamperes in sleep mode), when compared to other IEEE 802.15.4 transceiver/SoC solutions.

In Section 2.2, a study of a subset of wireless communication protocols is presented. The study focuses on communication protocols operating in the 2.4 GHz ISM band that mainly target factory automation applications, due to its more demanding timeliness and reliability nature. The presented review provides a generic introduction to each protocol followed by a brief analysis of its operation, emphasizing the adopted MAC mechanisms and discussing their performance and limitations.

The first observation is the existence of standard wireless protocols such as ISA SP100.11a [94], WirelessHART [98] and WIA-PA [103] that were specifically developed for factory automation applications and rely on the physical layer of the IEEE 802.15.4 standard for enabling communications. These protocols employ an OSI-like layer structure supporting a diagonal set of features that ranges from channel-hopping, for avoiding noise from co-located networks operating in the same band, to multi-hop communications, which allow extending the network coverage of the technologies. As noticed, these protocols employ a bandwidth reservation mechanism in the form of slot assignment scheme that allows ensuring deterministic timeliness to their transmissions. Nevertheless, given their higher complexity, multi-hop operation and particular configuration, the end-to-end delay experienced by these technologies can be high. For instance, in a simple star WirelessHART network with a 1500 milliseconds superframe duration, the average latency if of approximately 2 seconds with 92 milliseconds of jitter.

Another observation corresponds to the existence of wireless protocols designated specifically for a particular purpose, such as the "Wireless Fieldbus for Plastic Machineries" by Flammini *et al.* [93] or the WISA protocol by Dzung *et al.* [85], for example. Although less flexible and encompassing a single point of failure, these protocols are capable of meeting more stringent timeliness requirements due to their simplicity and adopted communication techniques. For instance, the worst case latency of the WISA protocol is bounded by a 15

milliseconds threshold.

The coexistence support and evaluation amongst protocols is highly heterogeneous, representing yet another observation from this study. Protocols such as WISA and WirelessHART provide both channel hopping mechanisms that allow to avoid noise from co-located networks operating in the same region of the spectrum, for example. Conversely, although protocols like the "Real-Time Sensor/Actuator Network for Factory Automation" provide mechanisms for channel redundancy and channel hopping, there is little or no information in the literature about their effective performance. Because the presented protocols operate in the 2.4 GHz ISM band they may become exposed to different types of noise from contending technologies. Since, for the most part, these protocols are based on the IEEE 802.15.4 physical layer and MAC, their resilience performance against noise from co-located network technologies is very important.

In general, the studied protocols assume a certain amount of free band in order to operate correctly. Although some of the studied protocols employ frequency agility mechanisms, these are only effective if there is enough bandwidth available to meet the application's requirements. When that is not the case, a significant performance degradation is to be expected, possibly rendering the network unusable. In many scenarios, this will result in considerable financial losses.

The analysis of relevant literature regarding the coexistence of the IEEE 802.15.4 technology in the 2.4 GHz ISM band is presented in Section 2.3. Provided the widespread adoption of technologies such as IEEE 802.11, IEEE 802.15.4 and Bluetooth, there is a non negligible probability that a deployed wireless real-time protocol may become exposed to them during its lifetime. This may occur due to either unintentional or intentional actions. The first occurs when an ad-hoc Bluetooth network is established in a factory setting to support some temporary file transmission or a Wi-Fi network is deployed without proper radio frequency planning, for example. Intentional actions occur when a third party controls the interfering networks to have some gain from his action. For example, a denial of service attack could make a wireless based production process to halt, benefiting the company's competitors. Hence, a study of the IEEE 802.15.4 protocol resilience to Wi-Fi, IEEE 802.15.4 and Bluetooth noise transmissions was performed.

Concerning the coexistence of co-located IEEE 802.15.4 networks, results indicate a significant degradation of the PER (above 60 %) when the SIR is small, as a consequence of high power interference transmissions from neighbor stations. Furthermore, it was noticed that, even transmissions on near channels (minimum offset) can have a visible effect (4.5 %)

on the PER of a IEEE 802.15.4 network, due to the spurious emissions that are generated in the sidelobes. The Bluetooth technology was found to have a smaller impact on the IEEE 802.15.4 PER (a maximum of 16 %). This is justified by both its frequency-hopping scheme, which oftentimes "jumps" between frequencies, and by its blacklisting mechanisms that, upon detecting channels with high PERs, marks and avoids them in the channel hopping pattern.

The coexistence of IEEE 802.15.4 with IEEE 802.11 networks was separately analyzed for scenarios encompassing IEEE 802.11b, 802.11g and 802.11n stations, due to their specificity. When compared to the IEEE 802.15.4 and Bluetooth technologies, a dramatic increase in the PER is observed for the three Wi-Fi versions of interference. As aforementioned, the SIR is tightly coupled with the PER. Results are reported with PER values above 50 % and, in some cases, going beyond the 90 % threshold. A PER increase resulting from Wi-Fi (IEEE 802.11g) sidelobe spurious emissions was also observed in this coexistence review. Furthermore, it was also found that, due to its wider bandwidth, transmissions conforming to the IEEE 802.11n standard have a higher impact on the performance parameters of the IEEE 802.15.4 protocol.

"A man gazing at the stars is proverbially at the mercy of the puddles in the road."

Alexander Smith (1830 - 1867)

3

Enforcing Traffic Separation in Open Environments

The previous chapter concluded that contention-based interference from moderate/high-power communication technologies can severely impair other on-going low-power communications in their range. This result challenges the support of real-time communications using low-power technologies in open environments, where such technologies are abundant and can be used without proper control. In the following sections, an overview of the *black-burst* contention method is provided and a novel medium access mechanism named *bandjacking* is proposed. This mechanism exploits the specific characteristics of contention-based communications to provide a deterministic channel access. Afterwards, the *bandjacking* reference architecture and operation are defined. Following, two different implementations of its key hardware element (the programmable interference synthesizer) are described and evaluated. Finally, the *bandjacking* effectiveness is assessed using a commercial off-the-shelf based programmable interference synthesizer.

3.1 Black-Burst Contention: An Overview

The idea of contending for channel access using jamming signals is not new. It was proposed more than ten years ago by Sobrinho and Krishnakumar [131], who defined a mechanism entitled "Ethernet Quality of Service Using Black Bursts" (EQuB). This mechanism adopted a Medium Access Control (MAC) overlay scheme, where real-time stations use jamming signals - called *black-bursts* - to gain prioritized channel access. EQuB was proposed under the assumption that real-time stations access the channel in (more-or-less) regular intervals and that network cards can detect packet collisions and send jamming

signals of pre-specified durations. The EQuB mechanism operates as follows. Real-time stations are expected to generate a continuous data stream over long periods of time, i.e., sessions. At the beginning of each session, a real-time station transmits the first packet following standard Ethernet rules. Down to the end of the session, subsequent packets are transmitted employing the scheme described in the next paragraph. When a real-time station successfully transmits a packet, it schedules a subsequent transmission for a given instant in the future.

Considering a scheduled access for the current time, if the real-time station finds the channel idle for an amount of time equal to the Inter-Frame Space (IFS), it starts the transmission immediately. Otherwise, it waits until this condition is met and initiates the packet transmission afterwards. Subsequently, if the real-time station detects a collision, it holds back the packet being sent and transmits a jamming signal (*black-burst*) in the channel. The *black-burst* length is proportional to the time that the station has been waiting for channel access, i.e., the time elapsing from the scheduled access attempt and the instant when the channel was perceived idle for an IFS. If, during the *black-burst* transmission, the real-time station senses that no collisions are occurring, it resumes the previous packet transmission that becomes successful. Hence, provided that only one station wins the contention in each *black-burst* window, the EQuB mechanism enables a round-robin service discipline among real-time stations.

The approach adopted for EQuB cannot be directly applied in standard wireless networks due to the lack of collision detection support. Consequently, Sobrinho and Krishnakumar proposed an extension of the EQuB mechanism for Wireless Local Area Networks (WLAN) [132]. This protocol differs from the latter in the sense that real-time nodes contend for channel access after the WLAN medium IFS instead of the long IFS, thus having a higher access priority than standard data stations. Besides, because real-time stations are not able to sense collisions, the *black-burst* cannot be shortened as in the EQuB when the channel is sensed idle, i.e., real-time stations transmit the *black-burst* for a period proportional to the time that the station has been waiting for channel access. Afterwards, real-time stations observe the channel for a given interval to determine, without ambiguity, if the station has the longest *black-burst* window, which enables the transmission of its data. In this case, the packet is transmitted and a new transmission is scheduled. Otherwise, as in EQuB, the real-time station will wait for the channel to become idle for a medium IFS interval. As before, this mechanism enables distributed prioritized access of real-time stations in contention-based networks.

Lindgren *et al.* [133] evaluated the provision of Quality of Service (QoS) in IEEE 802.11 wireless LANs with a study focused in four mechanisms: IEEE 802.11 Point Coordination Function (PCF), IEEE 802.11e Enhanced Distributed Coordination Function (EDCF), Distributed Fair Scheduling (DFS) and *black-burst*. The conclusion was that the *black-burst* mechanism exhibits the best performance for QoS support.

An enhanced version of the *black-burst* protocol was proposed by Jacob *et al.* in [134]. This protocol (enhanced *black-burst*) aims at improving the QoS assurance for Variable Bit Rate (VBR) traffic by allowing two contention mechanisms: one based on *black-bursts* and the other on EDCF. The authors propose a scheme where *black-burst* contention is used during the "bursty" periods of a VBR session and EDCF contention during the beginning of the bursts after the silence periods. This approach harnesses the benefits of both contention mechanisms, i.e., smaller access delays during VBR data bursts and after "long periods of silence". As a result, the performance in terms of packet delay, throughput and channel utilization is significantly improved.

Despite being capable of providing multiple Quality of Service (QoS) levels for real-time communications, the aforementioned MAC mechanisms cannot guarantee bounded access delays in open scenarios where different contention-based communication technologies may (co)exist. The following section addresses the operation of the *bandjacking* MAC scheme aimed at coping with this requirement.

3.2 Bandjacking: A Forceful MAC Technique

The requirement of a MAC mechanism allowing establishing a privileged access to the medium for wireless communications, even in environments affected by contention from different technologies, motivated the development of the *bandjacking* technique [135], which was inspired by two key observations. The first is how contention is handled in wild nature environments, where a resource can be disputed by several parties. For example, there is evidence that competition can increase resource use diversity within a natural population of three-spine sticklebacks (*Gasterosteus aculeatus*) [136]. This means that competition in nature can incite similar individuals to seek alternative resources for coping with their living requirements. Those who ensure the access to the available resources are more likely to prosper. Another example is the accepted theory that natural mammal competition for females was the evolutionary mechanism driving males to develop a larger body size and the development conspicuous weaponry such as horns [137]. In this perspective, larger

mammals with the best weaponry are more likely to reproduce and ensure their lineage. These are only two examples indicating that, when there is contention for a shared resource, nature favors those who evolve to use alternative resources or those who specialize to have better chances of accessing to the shared resource. In a sense, *bandjacking* is both an evolution of a technology to use an alternative resource that allows increasing its chances of getting the medium freed and a technological specialization to improve its probability of accessing the shared medium.

The second observation that inspired *bandjacking* was the acknowledgement that real-time events should always have a higher priority than any others. This implies that in a communication scenario where real-time and best-effort services are being provided, the former should always be serviced, even if it means to starve the latter. In this sense, it is acceptable that a real-time transmission may interrupt an ongoing best-effort transmission, since a failure in the services supported by the latter will not have a significant impact overall. The *bandjacking* technique was devised as a way of enabling guaranteed real-time transmissions in open (uncontrolled) environments. This technique allows a station to forcefully gain access to the communication channel, even if other contention-based technologies are simultaneously contending for the access to the medium. Therefore, *bandjacking* allows one station, herein designated as *critical station*, to conduct transmissions to one (or more) standard station(s) with deterministic medium access delay.

An intuitive explanation of how the *bandjacking* works is presented before providing its formal definition. The *bandjacking* technique defines that a critical data transmission, i.e., a transmission performed by a critical station, is preceded by a fixed length *black-burst* transmission [131] that ultimately jams all ongoing communications. If the jamming signal is long enough, all stations will eventually find the medium busy and postpone (backoff) their transmissions to a later time. Provided that the critical station is able to transmit a data packet immediately after the *black-burst* sequence, without having to enforce any Inter-Frame Space (IFS), it will gain prioritized medium access when compared to stations respecting the IFS or performing Clear Channel Assessments (CCA) before initiating a transmission. The approach of transmitting the critical packet immediately after the *black-burst* sequence resembles the mechanism proposed by Moraes and Vasques [138] in the h-BEB collision resolution algorithm for shared Ethernet networks.

The *bandjacking* is the sequence of transmissions encompassing the *black-burst* and the critical data packet. The prioritization of critical data transmissions will occur even if multiple technologies operate within the same geographical area, as long as the time

elapsing between the *black-burst* sequence and the critical data packet is shorter than the minimum IFS or CCA for those technologies. Additionally, it must be guaranteed that the critical data transmission is perceived as occupying the medium for the time of its duration.

3.2.1 *Bandjacking* Formal Definition

In this subsection the *bandjacking* MAC technique is formally defined and proved correct under a specific set of conditions.

Definition The amount of time elapsing from the end of a *black-burst* and the beginning of a critical contention-free data transmission is defined as the *capture interval* τ.

Lemma 3.2.1 *A contending communication station performing a clear channel assessment perceives the medium as busy if it detects a known signal pattern; a signal having one or more parameters above a given threshold; or a combination of both. Otherwise, the medium is declared idle.*

Proof Assume that technologies employing a contention mechanism use some derivation of the CSMA technique. Hence, the channel sampling activity which precedes any new transmission attempt is common to all contention based communication technologies. This procedure aims at evaluating the medium for inactivity. From this evaluation one of two results may occur: the medium is perceived as not being used or an ongoing transmission is detected. The detection of a busy medium is performed by identifying if the energy present in the medium is above a given threshold, if a signal with a given encoding is present in the medium or by a combination of both. These detection mechanisms are applied during the CSMA's clear channel assessment phase of both IEEE 802.15.4 [46] and IEEE 802.11 [64] technologies, for example.

Lemma 3.2.2 *Communication stations employing contention mechanisms to access the medium are hindered from performing transmissions if the medium is always perceived as occupied.*

Proof From Lemma 3.2.1, a contending communication station can only find the medium as either busy or idle after performing a clear channel assessment using a specific method. Therefore, before attempting a new transmission, contending stations must scan the medium for ongoing packets. If the medium scanning detects an ongoing transmission, the station

will postpone its transmission. Because the mechanism is only repeated for a limited number of attempts, the station attempting the new transmission will eventually drop it if the medium is always found busy with ongoing transmissions. Therefore, the station will be hindered from performing any transmissions as long as the medium is found occupied.

Definition A *critical station* is defined as a communication station that has the ability to synthesize *black-burst* interference, making co-located contending stations perceive the medium as busy, and also perform contention-free transmissions.

Lemma 3.2.3 *A critical station synthesizing* black-burst *interference for a period of time long enough will force all of the co-located contending stations to backoff from the medium.*

Proof Assume that $\Theta = \{\theta_{IEEE802.15.4}, \theta_{IEEE802.11}, ...\}$ represents the set of maximum transmission durations of a group of different technologies, where $\theta_{IEEE802.15.4}$ represents the maximum duration of an IEEE 802.15.4 packet and $\theta_{IEEE802.11}$ is the IEEE 802.11 packet maximum duration. Assume also that $\theta_{MAX} = \max(\Theta)$ corresponds to the maximum packet duration of all contending stations. From Lemma 3.2.2, a contention station will not be able to access the medium if it is perceived occupied. Therefore, if a station has the ability to make the medium be perceived as occupied for a period of time larger than θ_{MAX}, all co-located contending stations will become idle before the end of this period. This is supported by two facts. First, any contending station that might have initiated a transmission before the beginning of the synthesized interference will finish its transmission before the latter ends. Second, during the transmission of the synthesized interference any co-located contending station will find the medium busy.

Lemma 3.2.4 *A critical station that is capable of performing a data transmission masked as one from a neighbor "alien" contending station ensures that an ongoing critical transmission will not be corrupted by that station.*

Proof Assume that the critical station can generate a signal, herein designated as protective interference, which does not affect its data packets but makes co-located "alien" stations to perceive the medium as occupied. If the critical station has the ability to perform a data transmission and, simultaneously, synthesize protective interference, then according to Lemma 3.2.2, during these periods, "alien" stations perceive the medium as busy and will hold back from transmitting. In this sense, the protective interference masks the data transmission as one from an "alien" station.

Theorem 3.2.5 *A critical station synthesizing* black-burst *interference for a period of time* $t : t \geq \theta_{MAX}$ *can, subsequently, perform a contention-free data transmission, shielded by protective interference, without risking obstruction from co-located "alien" contending stations, as long as the capture interval is shorter than the minimum IFS or CCA durations of any contending technology.*

Proof Assume that $\Xi = \{\xi_{IEEE802.15.4}, \xi_{IEEE802.11}, ...\}$ represents the set of minimum IFS delays of a group co-located contending stations, where $\xi_{IEEE802.15.4}$ and $\xi_{IEEE802.11}$ are the minimum IFSs of the stations using the IEEE 802.15.4 and IEEE 802.11 technologies, respectively. Also assume that $\Lambda = \{\lambda_{IEEE802.15.4}, \lambda_{IEEE802.11}, ...\}$ expresses the set of minimum CCA durations for all of the co-located contending stations in which $\lambda_{IEEE802.15.4}$ and $\lambda_{IEEE802.11}$ designate the IEEE 802.15.4 and IEEE 802.11 minimum CCAs, respectively. From Lemma 3.2.3, if the *black-burst* transmission has a duration higher than θ_{MAX}, no contending station is transmitting at the end of the *black-burst*, i.e., the medium is clear from contending station transmissions. Consider that $\xi_{min} = \min(\Xi)$ and $\lambda_{min} = \min(\Lambda)$ define the minimum values of the IFS and CCA durations. If a critical station performs a contention-free data transmission immediately after the end of the *black-burst*, with a capture interval $\tau : \tau < \min\{\xi_{min}, \lambda_{min}\}$, it will not experience any obstruction from co-located contention stations, since they must respect their MAC technology requirements (minimum IFS or CCA durations) and will not initiate any transmissions during the capture interval. From Lemma 3.2.4, if the contention-free data transmission is accompanied by protective interference, then the co-located "alien" stations will be hindered from transmitting for the length of its duration, which ensures that the critical data transmission is not corrupted by them.

Building on the definition of the *bandjacking* MAC technique, the following subsections describe the reference architecture supporting its implementation and provide a detailed analysis of its operation.

3.2.2 Reference Architecture

The reference architecture describes the preferred method for implementing the *bandjacking* hardware and provides a network example which allows explaining the technique in a practical perspective. Figure 3.1 presents the architecture of a wireless network encompassing stations that employ different communication technologies and share the wireless 2.4 GHz ISM medium. For the purpose of better explaining the *bandjacking* technique,

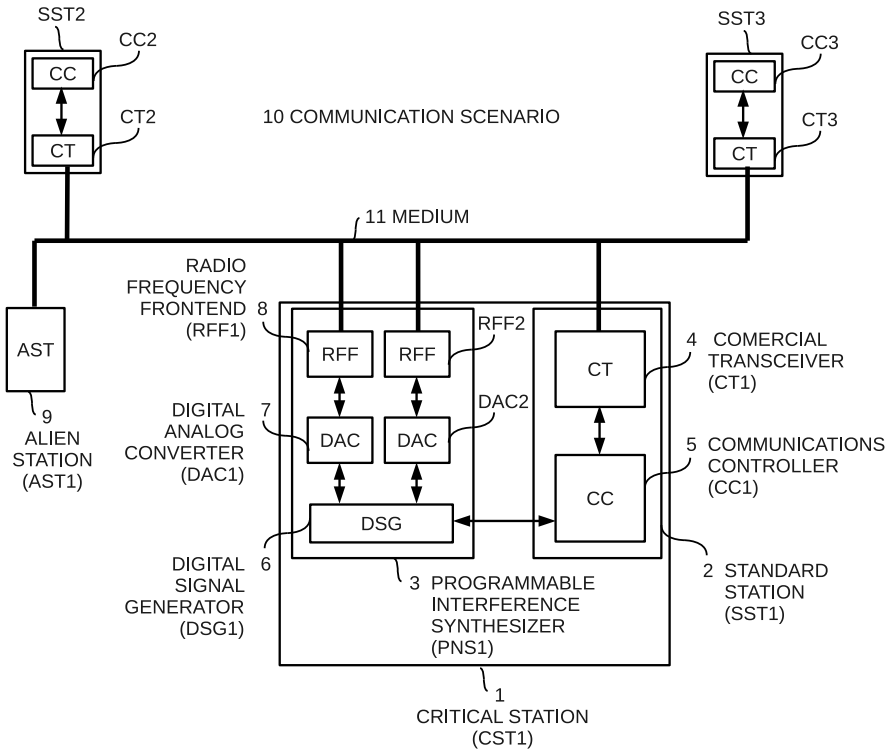


Figure 3.1: Architecture of a wireless network

without loss of generality, it is assumed that data communications are carried on with IEEE 802.15.4 transceivers and that transmissions from co-located IEEE 802.11 stations are perceived as interference. Since the latter stations are not part of the (IEEE 802.15.4) data network they are, herein, designated as "alien" stations.

The following considerations are valid for scenarios where multiple "alien" stations compete for the medium, despite the provided example encompasses a single "alien" station. In addition to the "alien" station (AST1), the network comprises one critical station (CST1) and two standard stations (SST2 and SST3). The critical station integrates a programmable interference synthesizer (PNS1) and a standard sub-station (SST1).

Standard stations encompass both a Commercial Transceiver (CT) and a Communications Controller (CC). The first one is responsible for transmitting/receiving data packets to/from the network while the second manages all the communications of the station. In the following discussion, without loss of generality, the CT is assumed to be an IEEE 802.15.4 compatible radio operating in channel 17. This channel covers a range of frequencies denoted as "data band" (21) in Figure 3.2. The notation (21) identifies a specific element in a figure unambiguously, in this case the element marked with the number 21.

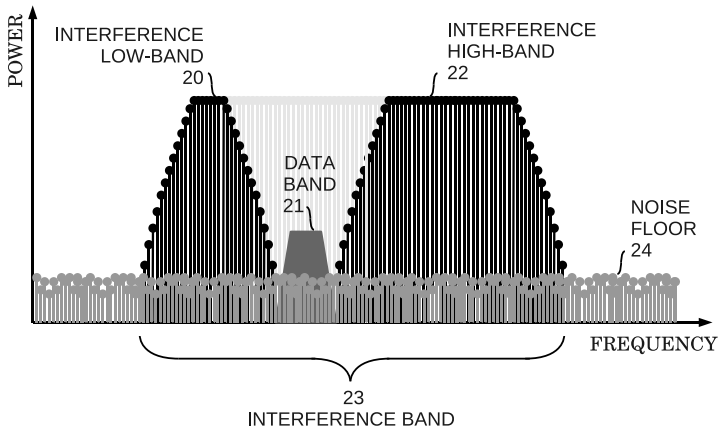


Figure 3.2: Spectrum occupation of the shared medium

This notation is adopted in the remainder of this section to reference the details in the figures more easily.

The CT can use one of the following methods to access the shared medium: (slotted or unslotted) Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) contention; contentionless Guaranteed Time Slots (GTSs); and contentionless direct access, a special access mode provided by some manufacturers such as Microchip. This mode allows a transceiver to initiate a transmission without waiting for any number of backoff slots or performing a Clear Channel Assessment (CCA).

The "alien" station (AST1) operates on the IEEE 802.11 channel 6, corresponding to the interference band (23) represented in Figure 3.2, and accesses the medium using the Distributed Coordination Function (DCF), which employs the CSMA/CA algorithm defined in the IEEE 802.11 specification [64]. This station is included in the example to represent the interference generated by an "alien" contention-based communication technology.

The programmable interference synthesizer (PNS1) produces interference profiles according to Figure 3.2, generating radio signals with a spectral content similar to the "alien" interference (23) or interference in both low (20) and high (22) sub-bands. This interference must not be perceived by standard stations as incoming packets, as they will attempt their decoding, which may affect their response time in case a transmission is required (e.g., immediately after the interference sequence). As it will be detailed further ahead, the requirement of performing simultaneous transmissions in both sub-bands aims at avoiding "alien" stations from finding the medium idle when a low-power critical data packet transmission is ongoing. All transmissions from the programmable interference synthesizer are conducted without contention.

Along with two front-ends (8) and two digital-analog converters (7), the PNS1 (3) also integrates a Digital Signal Generator (DSG) (6), which allows to simultaneously synthesize, for example, two different Direct Sequence Spread Spectrum (DSSS) encoded digital signals. Although we consider the generation of DSSS signals in this example, the DSG can be used to synthesize other signal types, as long as their characteristics are within the bounds of the applicable radio emission regulations. Furthermore, the DSG (6) also controls the local oscillator frequency used in the upconversion to the 2.4GHz ISM band as well as the gain of the RFF amplifier stage. In a critical station (1), all parameters related to the generation of interference (frequency band, power, duration) are managed by the communications controller (5) embodied on the standard sub-station (2).

Assuming that an IEEE 802.11b-like interference signal is to be synthesized by the PNS1 (3) according to the parameters passed on by the CC1 (5), the DSG produces a 11 Mbit DSSS encoded random signal, which is fed to one of the DACs, converted into analog, and passed to the following RFF (8). Here, the analog signal is up-converted, for example, to the central frequency corresponding to channel 6, and amplified with a gain that results in a transmission with 20 dBm of power (EIRP). Both RFFs shall be connected to omni-directional antennas. To synthesize interference simultaneously in the two sub-bands (20) and (22) a similar approach is followed by using two DSSS encoded random signals with appropriate bandwidths, but guaranteeing that the cross-interference in the data band is bellow its noise floor level. The synthesis of this interference profile must be carefully implemented, as non-negligible problems to the integrated low-cost transceivers operating in the data band (21) may occur, for example, due to saturation phenomena.

The legal limit for the transmission of signals in the 2.4 GHz ISM band is of 100 mW (20 dBm) of Effective Isotropic Irradiated Power (EIRP) in Europe (ETSI EN 300 328) and 1 W (30 dBm) of Peak Conducted Output Power in the US (FCC part 15.247 and 15.249), assuming the transmission of Direct Sequence Spread Spectrum (DSSS) signals. Therefore, the synthesized interference in the (20), (22) and (23) bands must comply with these limits.

3.2.3 Operation

Figure 3.3 represents a set of signals associated with the communication scenario represented in Figure 3.1. Assume that a standard station SST2 has been previously attached to a neighbor Wireless Personal Area Network (WPAN) coordinator (not shown) operating in the beaconed mode and that it was granted a GTS to transmit a data packet in a given

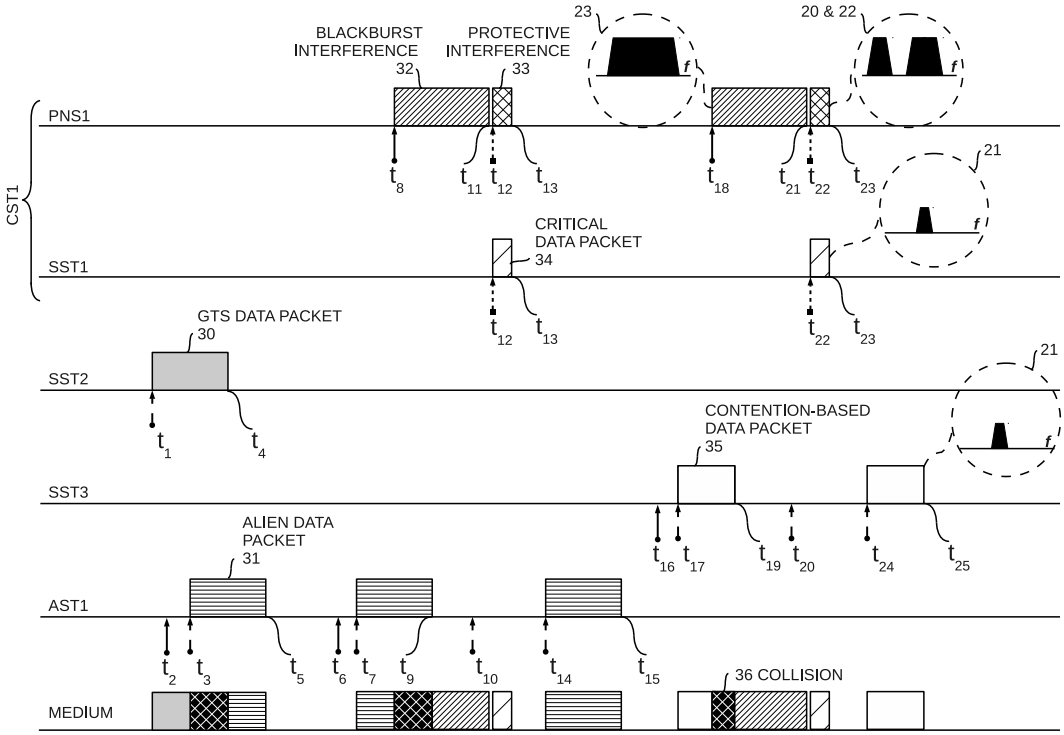


Figure 3.3: Timeline of two *bandjacking* accesses

Contention Free Period (CFP). Suppose that a GTS data transmission (30) is initiated at instant t_1 and finished in t_4 by the SST2 station. An "alien" station (AST1) becomes ready for initiating a data transmission at instant t_2. However, it must observe the medium for a period of time known as DCF Inter Frame Space (DIFS) before declaring it inactive at instant t_3 and proceeding with an "alien" packet transmission (31) until the t_5 instant.

As it is shown, a collision takes place in the medium from t_3 to t_4. This may occur since the transmissions performed by short-range wireless communication technologies (e.g. IEEE 802.15.4) may not be detected by co-located technologies employing higher levels of power in their transmissions (e.g IEEE 802.11). In other words, the energy levels produced by low-range transmissions may not be strong enough to make "alien" stations declare the medium occupied. As such, all standard low-range communications, regardless of the medium access method, are prone to interference from other contention-based networks, especially those employing higher power transmissions.

Assume that the "alien" station (AST1) becomes ready for an "alien" transmission (31) at instant t_6 and starts it at instant t_7, after the medium has been observed for the DIFS period. This transmission is concluded at instant t_9. If the critical station (CST1) has a data packet scheduled for instant t_{12}, it starts the transmission of a *black-burst*

interference sequence (32) at instant t_8 with a spectral content (23) similar to that of the "alien" technology, jamming the "alien" packet transmission. The interference sequence is completed at instant t_{11}. Immediately after, at instant t_{12}, the scheduled critical data packet (34) is sent in the data band (21) by the standard sub-station SST1, finding the medium free and ending at instant t_{13}. During this period, two "protective" interference signals are also simultaneously produced by the synthesizer in the lower (20) and upper (22) sub-bands, as depicted in Figure 3.2 and 3.3.

The need to perform the data transmission immediately after the interference sequence is a consequence of the Theorem 3.2.5, which implies that the time window elapsing between the events t_{11} and t_{12} must be smaller than the IFS and CCA of any neighbor "alien" technology. This guarantees that, even if an "alien" station initiates the CCA during this period, it will acknowledge the channel as being occupied because it will find significant energy levels. This occurs due to either the subsequent data transmission (34) or the simultaneous transmission of interference (33) in the lower (20) and upper (22) sub-bands. The same is valid for an "alien" station waiting the IFS before initiating a transmission.

The *black-burst* interference sequence (32) ensures the medium availability at instant t_{12} by inhibiting "alien" stations from starting transmissions during this period. The protective interference sequence (33) ensures that "alien" stations will not mark the medium as free when, for example, low-power narrow-band data transmissions are on-going. Provided that the critical data is transmitted simultaneously to the protective interference (33) immediately after the end of the *black-burst* interference sequence (32), "alien" DoS attacks using (contention-based) packet flooding will not be effective in blocking the critical transmissions.

According to Figure 3.3, after failing the transmission initiated at instant t_7, the "alien" station AST1 tries to access the medium at instant t_{10}, but finds it occupied and performs a backoff procedure until instant t_{14}, where it acknowledges the medium as free and proceeds with a successful transmission lasting until instant t_{15}.

A similar cycle is initiated at instant t_{18} corresponding to the dispatch of a critical data packet. Another interference sequence (32) is initiated even with the medium occupied by a contention-based data transmission (35) triggered by the standard station SST3 at instant t_{17}. In this scenario, the *black-burst* interference sequence (32) between instants t_{18} and t_{21}, the critical data packet (34) between instants t_{22} and t_{23}, and the simultaneous protective interference (33) repeat. As illustrated, the ongoing data transmission (35) is corrupted by the interference sequence (32) between instants t_{18} and t_{19}. Therefore, station

SST3 evaluates the medium at instant t_{20} to attempt a packet retransmission, finding it occupied. After conducting the backoff procedure, station SST3 evaluates the medium at instant t_{24} and, observing it free, initiates a packet retransmission lasting until instant t_{25}.

3.3 SDR-based PNS Implementation

The first hardware solution for the Programmable Interference Sequencer (PNS) is described in this section. This version of the PNS was devised in collaboration with the Wireless Sensor Networks LABoratory Group of the Electronics for Automation Department of the University of Brescia, which was responsible for implementing the software running in the PNS. Although the initial approach was aimed at implementing a hardware platform similar to the one defined for the PNS in Figure 3.1, soon it was realized the availability of Software Defined Radio (SDR) development KITs with a similar architecture. In this sense, an option was made for evaluating the feasibility of using one SDR KIT as the PNS. The remaining of this section describes the SDR-based PNS architecture and operation with an emphasis on the solution's hardware and software. In addition to providing an in-depth review of the PNS implementation, this section also documents the testbed used to assess its performance and discusses the obtained results with the perspective of building a SDR-based commercial solution for the PNS.

3.3.1 Architecture

The use of a SDR-based solution for the PNS was initially motivated by the similarity between the hardware architectures of both the envisaged PNS and of a typical SDR. This option was also supported by two other advantages. First, the flexibility and reconfigurability offered by this approach allows a rapid modification of the signals' characteristics without depending on hardware redesign. Second, the openness provided by the joint use of the GNU Radio open-source software development toolkit and of the Universal Software Radio Peripheral (USRP) platform makes the development of radio solutions more agile due to software component reuse.

Hardware

Figure 3.4 shows the typical block diagram of a SDR platform. This diagram emphasizes the abstract blocks of the SDR and the representative physical components contained

in each one. As documented, there are three main blocks: *Data Processing*, *Wideband Conversion* and the *Radio Frequency Front-end*. The data processing block, as the name suggests, is responsible for the SDR's data processing abilities. It can be implemented by means of a generic processing unit that encodes/decodes digital samples. The implementation of this block usually relies on a DSP; a microcontroller; a FPGA; a general purpose PC; or a combination of these. An effective approach is to have the data processing block implemented using an hybrid structure, where high speed repetitive tasks are ensured by programmable logic (e.g., CPLD or FPGA) and tasks with a higher level of abstraction are handled by a dedicated processor.

The wideband conversion block separates the analog and digital sections of the SDR. This block typically defines the analog bandwidth of the system and is commonly the SDR's bandwidth bottleneck. As shown in Figure 3.4, it holds both the ADC and the DAC, which are responsible for digitizing the analog signal coming from the Radio Frequency Front-end (RFF) and converting to analog the digital signal synthesized at the data processing block, respectively.

Finally, the RFF is responsible for tuning the analog baseband signal coming from the wideband conversion block to be wirelessly transmitted and to transform the radio signals coming from the antenna to be digitized by the DAC. The transmission path of the RFF is generally composed by an Intermediate Frequency to Radio Frequency (IF $\rightarrow$ RF) stage and a Power Amplifier (PA) just before the antenna. The former stage performs the up-conversion that translates the software-configurable IF into the RF of interest. The later increases the amplitude of the signal fed to the antenna for an increased communication range. The receiving path includes stages with complementary functions: a Low Noise Amplifier (LNA) and a (RF $\rightarrow$ IF) stage. The first, while maintaining the levels of noise to a minimum, produces an amplified version of the signal received at the SDR's antenna. The second performs a down-conversion that translates the center frequency of the RF signal to an intermediate frequency suited to be handled by the wideband conversion block.

The SDR concept can be developed in the 2.4 GHz ISM band using several commercially available tools, as presented in Table 3.1. For example, National Instruments provides the USRP-2921 [139] for 2380 €. This software radio bundle allows synthesizing signals covering the 2.4 GHz and 5 GHz ISM bands, thus supporting Bluetooth, Wi-Fi and ZigBee communications. This KIT allows interfacing with LabVIEW for streaming baseband IQ signals with up to 25 MS/s. LabVIEW is an application that enables the development of algorithms for physical layer communications using an intuitive graphical programming

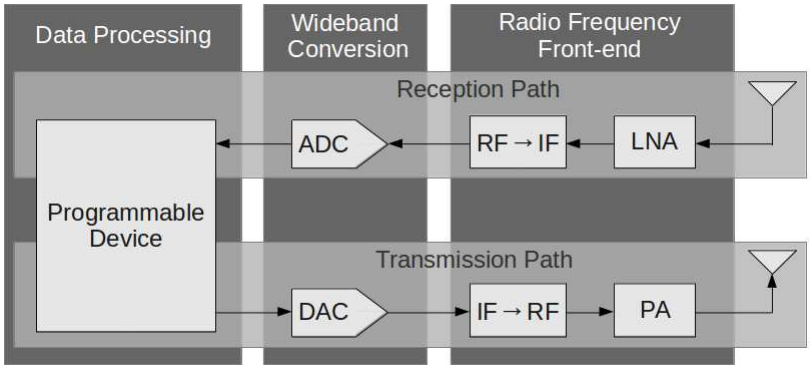


Figure 3.4: Block diagram (top) and physical component mapping (bottom) of an SDR

Table 3.1: Commercial SDR KITs

Vendor	Designation	Frequency range (MHz)	Cost (€)	Software support
National Instruments	USRP-2921	2400-2500, 5725-5875	2380	LabVIEW
Avnet	ZYNQ SDR-II	70 -6000	≈ 1100	Xilinx Vivado
Great Scott Gadgets	HackRF Jawbreaker	902-928, 2400-2500	≈ 222	GNU Radio
Ettus Research	USRP1 + RFX2400	2300 - 2900	≈ 860	GNU Radio

approach that increases productivity.

Avnet is another SDR provider. This manufacturer commercializes the ZedBoard ZYNQ SDR-II [140] KIT for approximately 1100 €. This software defined radio KIT combines a ZedBoard with a Analog Devices AD-FMCOMMS2-EBZ FMC module, featuring the AD9361 integrated RF agile transceiver. Although the transceiver can be tuned for RF center-frequencies ranging from 70 MHz to 6 GHz, it is optimized for synthesizing signals in the 2400 - 2500 MHz region of the spectrum with a channel bandwidth ranging from 200 kHz to 56 MHz. The development of communication solutions using the ZedBoard is conducted with the Xilinx Vivado Design Suite, which enables key productivity elements such as IP integration and implementation; verification and debug; and design exploration and IP generation.

The HackRF open source project [141] is a SDR platform that is gaining a considerable attention in both the academy and the industry. This project aims at producing a SDR device that can operate over a wide range of frequencies, including both the 2.4 GHz and 900 MHz ISM bands. This project was funded for more than 444000 € in Kickstarter. The first release of the HackRF board was named Jawbreaker and became available in January 2014, with a cost of around 222 € per board.

The most popular platform for developing the SDR concept is the Universal Software

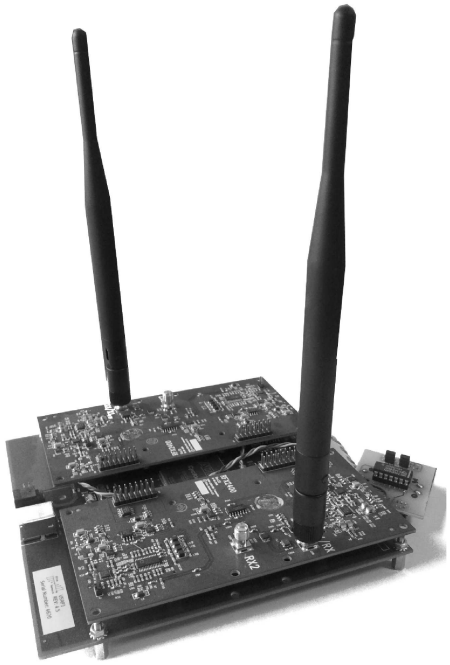


Figure 3.5: USRP-based PNS implementation and μMRF mezzanine interface board

Radio Peripheral (USRP) created by Ettus Research, LLC [142]. The first SDR system was the USRP1 that, when paired to the RFX2400 daughterboard, is capable of synthesizing signals occupying the spectrum region between 2300 MHz and 2900 MHz with host signals with a maximum bandwidth of 16 MS/s. This device is a low-cost and high-speed equipment designed to "allow general purpose computers to function as high bandwidth software radios" [143]. The USRP1 board is available together with the RFX2400 daughterboard for approximately 615 € and 245 €, respectively. Over the last years, Ettus Research has broadened its SDR offer in three product categories: USRP Networked Series, USRP Embedded Series and USRP Bus Series. The Networked Series targets high-bandwidth and high-dynamic range applications that require Gigabit Ethernet connectivity. The Embedded Series, as the name suggests, addresses embedded applications where the use of a computer together with the USRP is not desired. This platform combines the flexibility of the USRP product family with an embedded processor running a custom Linux image. Finally, the Bus Series extends the USRP1 line by including both fully integrated, single-board USRP solutions as well as USRP Kits that mount daughterboards like the USRP1. The USRP platform is commonly paired with the GNU Radio software suite to rapidly create complex SDR systems without the burden of developing commonly used software defined building blocks from scratch.

The adopted SDR platform for developing a 2.4 GHz ISM band programmable interference synthesizer (PNS) was the USRP1 Kit together with RFX2400 daughterboard by Ettus Research LLC. This option was motivated by its flexibility, low cost and broad open source support, namely by the GNU Radio suite and its UCLA Zigbee block. The USRP1 motherboard hosts two 12-bit ADCs (up to 64 MS/s sampling rate), two 14-bit DACs (up to 128 MS/s) and an Altera Cyclone EP1C12 FPGA for simple high speed operations such as up-conversion, down-conversion, interpolation and decimation. The ADCs and DACs allow both the reception of baseband signals of up to 32 MHz of bandwidth and the generation of baseband signals of up to 50 MHz of bandwidth. To ensure the ability to tune two different frequency ranges, a set of two RFX2400 daughterboards were mounted on the USRP1 board. As depicted in Figure 3.5, a μMRF mezzanine board was added to allow the critical station's communications controller to manage the transmission of *black-burst* and protective interference. This board was designed to individually control each of the RFX2400 daughterboards by applying a control pulse to the ENOP pin of the Analog Devices' AD8349 quadrature modulator chip on each board, thus allowing to switch it on and off within 50 nanoseconds and, therefore, start/stop the transmission of any signal being conveyed by the USRP1 board.

Software

The GNU Radio [144] open source project was elected for the implementation of the PNS software architecture due to its support for developing "*real-time, high-throughput radio systems in a simple-to-use, rapid-application-development environment*" [144]. The GNU Radio framework is composed by a modular, expandable and block-organized software library, which can be further enriched with signal processing blocks contributed by the GNU Radio community. The GNU Radio platform is being used in both research and commercial domains for developing wireless communication solutions by hobbyists, academics and enterprises.

The software structure of a GNU Radio application is mainly written with the Python programming language, a dynamic object-oriented scripting (interpreted) language. However, the performance-critical elements of the signal processing path are usually implemented in C++, taking advantage of any available specific processor floating point extensions. The integration between C++ and Python occurs by means of the Simplified Wrapper and Interface Generator (SWIG), an interface compiler, allowing data to flow at maximum speed.

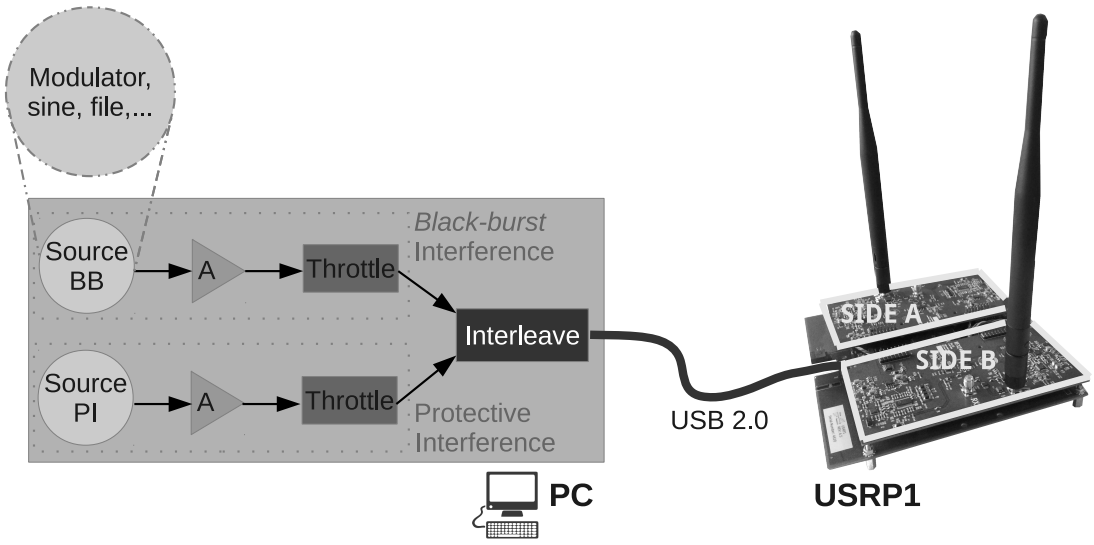


Figure 3.6: Block diagram of the PNS prototype using the GNU Radio framework

Figure 3.6 shows the software block diagram of the implemented PNS. As it can be seen, two different paths are represented: one for the *black-burst* interference and the other for the protective interference. In both paths there is a data source, an amplifier and a throttle. The two signal flows are then interleaved and sent to the USRP using a USB 2.0 connection. These two interference signals are then demultiplexed at the USRP and fed to an individual RFX2400 daughterboard, allowing them to be switched on and off separately. The source block works as the baseband sample generator and can be replaced by a generic modulator or a file source, for example. Since the original aim of this implementation was to create a PNS prototype, the signal source block was left undefined. Provided that it is a software block, it can be easily replaced (via software reprogramming) with what best suits the needs of testing, such as a particular modulator, a signal generator function or a (pre)recorded file.

Following the signal path, after the sample generator block (source), a digital gain amplifier is placed in order to amplify the source signal amplitude, which is of ± 1 (floating point value). After passing the gain block, the signal increases to an amplitude of ± 8000 points (integer value). The selected gain was chosen because it represents a good compromise for a low Spurious-Free Dynamic Range (SFDR) with the USRP [145]. The amplified signal then enters the throttle block, which moderates the rate of the streaming data to avoid overloading the CPU. The block output is a flow of samples in which the average rate does not exceed a given defined value of samples per second. In this case the selected value is 4 MS/s. The two data flows encoding interference signals are then merged by the

interleave block and then sent to the USRP via USB 2.0 connection.

One of the main advantages of developing a SDR-based PNS is the ability to change the characteristics of the interference signals simply by modifying the data source and the configurations of the front-ends.This possibility prompted the endeavor of trying different approaches for finding a suitable interference pattern, hindering IEEE 802.11 stations from transmitting when other IEEE 802.15.4 transmissions are both beginning or ongoing.

Boano et al. [146] found that a signal synthesized by a USRP board with enough power can completely block IEEE 802.15.4 communications. Furthermore, in a different approach, several studies indicate that IEEE 802.15.4 transmissions have significant impact on IEEE 802.11 communications. For example, in [147], the PER of the IEEE 802.11b technology is analyzed under the interference of IEEE 802.15.4 transmissions. The authors conclude that if the distance between both transceivers is less than 2 meters the interference caused by the IEEE 802.15.4 transmissions on IEEE 802.11b communications causes a PER of around 0.8 (80% of all packets transmitted by the IEEE 802.11b stations are lost). In a more recent work by Mao et al. [148], the coexistence of the IEEE 802.15.4 and IEEE 802.11b/g technologies is studied by means of analyzing the throughput of IEEE 802.11b/g communications under the influence of IEEE 802.15.4 transmissions. One conclusion is that the throughput drops to about 59% when the IEEE 802.15.4 station is at one meter from the IEEE 802.11b/g station. Despite increasing the distance between these stations has, generally, a positive effect on the throughput, that does not always occur due to signal multi-path phenomena. In both cases, results indicate that IEEE 802.15.4 signals have enough bandwidth to be detected by IEEE 802.11 stations. Nevertheless, in our approach to implement the PNS, two different scenarios were considered: protective interference generated across multiple IEEE 802.15.4 channels or (*black-burst*) interference on a single channel. In both cases an IEEE 802.15.4 modulator was used as the source signal for the PNS. Besides the potential capability of blocking IEEE 802.11 transmissions, the use of an IEEE 802.15.4 modulator was also motivated by its low implementation complexity, which comes from using the GNU Radio project [149].

The first scenario assumes that the interference signal must occupy more than one IEEE 802.15.4 channel to be effectively detected by neighboring IEEE 802.11b/g stations. Hence, a frequency hopping procedure was implemented, allowing the interference signal to hop over a configurable number of IEEE 802.15.4 channels. In the second scenario, it is assumed that the interference on a single IEEE 802.15.4 is sufficient for making IEEE 802.11b/g stations perceive the medium as busy.

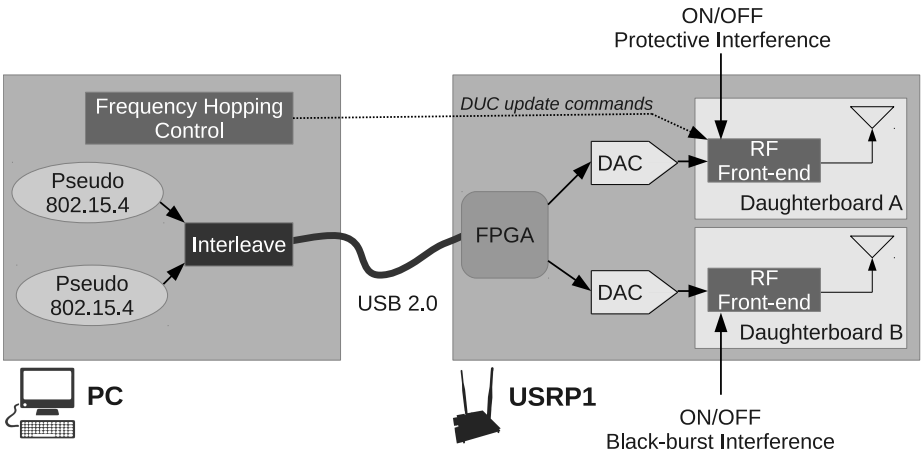


Figure 3.7: PNS block diagram: frequency hopping protective interference

In both scenarios, a stream of samples was generated and recorded into a file using the IEEE 802.15.4 modulator. The stored samples were processed to avoid being perceived as IEEE 802.15.4 valid data. This was conducted by changing the packet synchronization header, which allows standard IEEE 802.15.4 stations to ignore this signal, while keeping the original IEEE 802.15.4 band occupation. Changing the synchronization header is critical in scenarios when it is required that an IEEE 802.15.4 station responds immediately after the end of an interference sequence, such as the case of a critical transmission. If an IEEE 802.15.4 packet is perceived as valid, the transceiver will attempt its decoding, blocking the possibility of an immediate response.

As illustrated in both Figures 3.7 and 3.8, the block that reads the samples from the file and sends them continuously in a loop is represented as the pseudo 802.15.4 source block. Two pseudo 802.15.4 signals are generated: one to be used as *black-burst* interference and the other to be employed as protective interference. As described, these signals are then interleaved and sent to the USRP, which proceeds to its demultiplexing and feeding to the designated DACs. The interleaving is required to allow transmitting simultaneously both data flows over the single USB connection.

In the first implementation scenario, as depicted in Figure 3.7, a Frequency Hopping Control block is added to the transmission path of the protective interference side to allow occupying more than one IEEE 802.15.4 channel by means of frequency hopping. This block has the ability of sending update commands to the digital up conversion section of the USRP, changing the tuning frequency of the analog front-end after a given period of time. This block is not required in the fixed frequency scenario because the protective

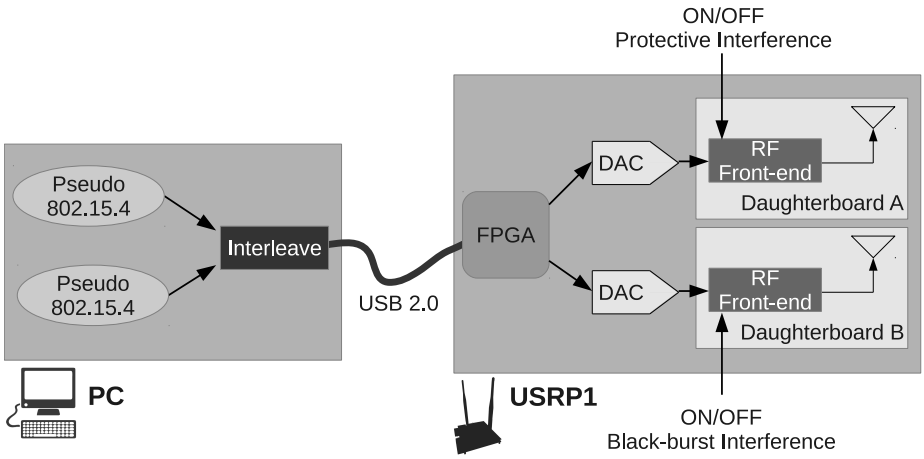


Figure 3.8: PNS block diagram: fixed channel protective interference

interference is transmitted on a single pre-defined channel, as shown in 3.8.

In both block diagrams, the FPGA demultiplexer splits the received signal stream in two flows; each one fed to a different radio frequency font-end and tuned to a specific center frequency (or set of frequencies) to match the required band occupation for each type of interference. Provided that the 2.4 GHz ISM band has been chosen to implement the PNS prototype, two RFX2400 daughterboards have been mounted on the USRP board, allowing tuning two different individual signals in this band of frequencies. These daughterboards host two radio frequency analog mixers that allow adjusting their operation in the 2400 to 2900 MHz range of frequencies. As documented in both Figures 3.7 and 3.8, the RF front-end of the daughterboard A is tuned for the channel (or set of channels) corresponding to the protective interference, while the front-end of the daughterboard B was configured to transmit in the *black-burst* interference channel, i.e., the same that is used for valid IEEE 802.15.4 data transmissions. The daughterboards permit that both types of interference can be individually turned on and off using external signals.

3.3.2 Evaluation

After implementing the PNS using a SDR-based approach follows the assessment of its effectiveness in blocking IEEE 802.11 transmissions. Of particular interest is experimentally finding the optimal gain/transmission power and the channel separation that enables the support of "protective" interference without affecting the critical data being simultaneously transmitted. In the coming section, besides presenting the methodology used to perform the PNS evaluation, the associated results and the key findings are provided.

Methodology

Figure 3.9 shows the test setup created to assess the effectiveness of the (USRP-based) PNS in hindering Wi-Fi communications. The testbed is composed by several elements, including:

- An IEEE 802.11b network between a laptop and a personal computer using an ad-hoc connection to emulate a noisy Wi-Fi environment. The laptop generates UDP traffic using the Distributed Internet Traffic Generator (D-ITG) application [150];
- A python script driving the PNS USRP board and running in the PC. The script generates a pseudo IEEE 802.15.4 data stream using packets with a payload of 9 bytes that, after being fed to the radio frequency front-end and tuned to a given IEEE 802.15.4 channel, will be interpreted as "interference" by neighbor Wi-Fi stations;
- A Wireshark application [151] is used in the PC to monitor the packets received by the attached Wi-Fi adapter in order to evaluate the Wi-Fi packet error impact of the pseudo IEEE 802.15.4 "interference";
- Two IEEE 802.15.4 stations (transmitter and receiver) are setup to transmit and receive packets to evaluate the ability of the pseudo IEEE 802.15.4 "interference" to protect standard IEEE 802.15.4 transmissions. The IEEE 802.15.4 stations run a specific firmware that allows to start/stop a trial and register the number of packets transmitted and received during that trial.

The two computers and the two IEEE 802.15.4 stations were placed over a table in the vertexes of a virtual square, as represented in Figure 3.9. The table has a height of 75 centimeters and the virtual square has a side length of approximately 1 meter. The USRP board of the PNS was placed on the same table, one meter apart from the wireless interface card of the PC and at approximately two meters from the laptop. The room where the trials were performed has an area of approximately 40 m^2 and was clear from interference beyond the one produced by the PNS.

Two evaluations were conducted. The first is related to the assessment of the frequency hoping version of the PNS, where several channels are used to transmit a pseudo IEEE 802.15.4 interference signal in a round-robin fashion. The setup shown in Figure 3.9 is used without the standard IEEE 802.15.4 stations in this evaluation. Hence, in order to evaluate how Wi-Fi transmissions are affected by the interference generated by the PNS,

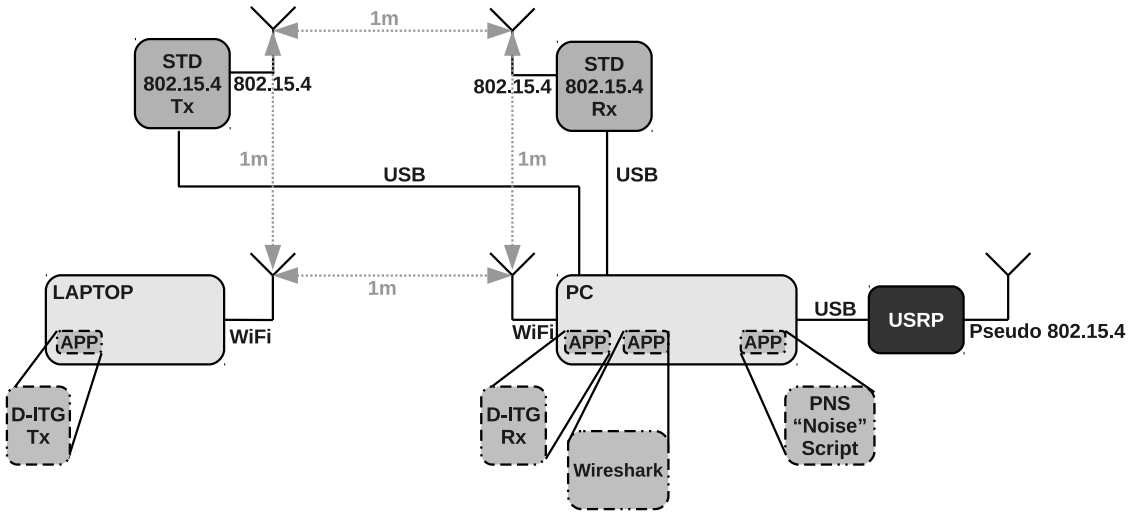


Figure 3.9: Testbed used to evaluate the SDR-based PNS

a Wi-Fi connection is established between the laptop and the PC. Using this setup, two sets of trials are conducted, including the transmission of UDP packets over the Wi-Fi link with:

- A payload of 1 byte and a period of 50 μs;
- A payload of 1458 bytes and a period of 1250 μs.

The motivation for using these packet lengths was to observe if their size has any influence on the ability of the PNS to block their transmissions. Furthermore, a set of trials encompasses four trials, each one having the PNS configured to perform successive pseudo IEEE 802.15.4 "interference" transmissions over a different set of channels. Because the Wi-Fi connection was established in channel 1 and the assessment aims at evaluating the Wi-Fi resilience to the PNS synthesized interference, the trials were performed using interference transmission on the overlapping 802.15.4 channels 11, 12, 13 and 14. In each trial, a different set of channels is employed in the round-robin channel hopping sequence. The first trial is conducted with pseudo IEEE 802.15.4 interference transmissions on channel 12; the second trial on channels 12 and 13; the third trial on channels 11, 12 and 13; and the fourth trial with interference transmissions on channels 11, 12, 13 and 14. In this evaluation, the PNS was setup in all trials with the following default configuration options: tune delay= 0.001 s, dwell = 0.001 s and gain = 8000. The tune delay corresponds to the time interval enforced between the tuning of a channel and the instant where the channel becomes available for use, i.e, it defines the period of time used to allow the radio frequency

circuitry to stabilize on the tuned channel. The dwell delay is the time interval that must elapse in each tuned channel before switching to a new one. Finally, the gain defines the amplitude of the output signal.

The second evaluation addresses the task of finding the required channel separation, which guarantees that the pseudo IEEE 802.15.4 interference produced by the PNS can be used as "protective" interference, i.e., as interference that can be sent simultaneously to the critical data, but on a IEEE 802.15.4 lateral channel. This evaluation is related to the fixed channel version of the PNS. The complete setup illustrated in Figure 3.9 is employed in this evaluation. As in the fist evaluation, the UDP traffic sent over the Wi-Fi connection was configured with payloads of either 1 or 1458 bytes in channel 1. The standard IEEE 802.15.4 stations were placed 1 meter apart and setup to communicate in channel 14 with both a power of -10 dBm and a period of 100 milliseconds. The PNS protection interference was synthesized in one of the channels 11, 12, or 13 with a configurable gain of 8000, 16000, 24000 or 32000. Each trial encompasses the transmission of 1000 IEEE 802.15.4 broadcast packets by a standard IEEE 802.15.4 device (transmitter) simultaneously to both the Wi-Fi traffic and the protective interference. The IEEE 802.15.4 standard receiver device registers which transmitted packets have been received and provides the number of packet errors in each case.

The following subsection provides the results collected in both evaluations and a discusses their implications.

Results

The frequency hoping PNS evaluation results are presented in Table 3.2. This table shows the number of Wi-Fi packets received by the Wireshark application when the Wi-Fi link is subject to four different channel interference sequences synthesized by the PNS. As depicted, no packets were received by the Wireshark application in any of the trials, which indicates that using the USRP to transmit pseudo 802.15.4 packet is a feasible way of synthesizing interference to inhibit Wi-Fi communications. Furthermore, it is observed that the use of a single IEEE 802.15.4 channel (12) is sufficient to entirely block the Wi-Fi communications. The use of different payloads for the UDP traffic sent over Wi-Fi links seems to have no effect on the pseudo 802.15.4 interference blocking capability.

Table 3.3 documents the results of the second evaluation, where a set of trials was conducted to find the best channel separation allowing the PNS to synthesize protective interference without risking corrupting the critical data communications. As demonstrated,

Table 3.2: Received Wi-Fi packets in the presence of synthesized interference

"Interference" channel(s)		12	12,13	11,12,13	11,12,13,14
UDP Length	1	0	0	0	0
	1458	0	0	0	0

Table 3.3: Packet errors in the presence of "protective" interference

Gain		8000		16000		24000		32000	
UDP Length		1458	1	1458	1	1458	1	1458	1
Channel	11	0	0	1	0	1	0	0	0
	12	0	0	45	0	5	258	16	113
	13	923	909	999	998	1000	997	1000	1000

the transmission of protective interference in channel 13 severely impairs the standard IEEE 802.15.4 data communications performed on channel 14. However, this effect is not observed when channels with a higher frequency separation (for a gain of 8000) are used. In addition, results show that the adoption of higher power protective transmissions, such as in channel 12 with gains of 16000, 24000 and 32000, causes an increase in packet errors in channel 14. The use of different payloads in the Wi-Fi traffic seems to have no clear tendency in affecting the protective interference effectiveness. Therefore, in order to guarantee that data transmissions are not affected by interference, regardless of its transmission power, a separation of two channels between critical data and protective interference is required.

As aforementioned, the use of a SDR-based PNS has the key advantage of allowing to experiment different signals and processing algorithms without changing the supporting hardware. However, this advantage comes with some costs. First, due to their specific purpose, SDR-based solutions are still not yet broadly adopted, as they are more expensive than equivalent commercial systems. Second, due to the flexibility requirement, they tend to be rather larger than custom made communication boards. Third, they usually require a PC or a high performance processor that handles the upper layer programs, which not only increases the overall cost, but also the power consumption. Finally, although the development process is usually simpler, it carries an increased debugging complexity due to the amount of code layers that must be analyzed. For these reasons, a low-cost PNS version was devised using a commercial transceiver, employing the knowledge gathered from evaluating the SDR-based PNS. In the following section, this low-cost PNS version is described and evaluated.

3.4 COTS-based PNS Implementation

As presented, the first attempt at developing a fully functional PNS adopted a SDR approach. In this endeavor, several general conclusions were drawn from the ability of blocking Wi-Fi transmissions using modified IEEE 802.15.4 transmissions, mainly concerning the channel selection. This information was crucial for the development of a low cost alternative to the SDR-based PNS.

In [146], Boano et al. propose and assess an IEEE 802.15.4 based low-cost interference generator allowing to audit WSNs operating in the 2.4 GHz ISM band. This tool generates interference using a Chipcon CC2420 transceiver [152] operating in two specific test modes, which allow synthesizing a unmodulated signal characterized by a highly concentrated power spectrum peaking at the center frequency of the selected channel and a randomly-modulated signal, which has its spectrum power spread evenly across the bandwidth of the selected channel. In the first case, the unmodulated carrier does not trigger packet handling interrupts in neighboring IEEE 802.15.4 stations. Conversely, because the modulated version employs a synchronization header that makes all IEEE 802.15.4 stations in range to actively receive the generated packets, these events are signaled by interrupts. The first mode of operation cannot be used in the development of commercial applications, since it only envisages testing procedures and does not comply with the radio emission regulations of the 2.4 GHz ISM band.

As defined in the previous section, in order to guarantee that an IEEE 802.15.4 station is able to respond immediately after the end of an interference sequence, the generated interference cannot be perceived as including valid IEEE 802.15.4 packets, since that would lead to an attempt of decoding them at the transceiver, which would impair the possibility of an immediate response from the IEEE 802.15.4 station. Provided that only the Chipcon's CC2420 second mode of operation can be adopted for developing a commercially viable solution and since its transmissions are perceived as valid IEEE 802.15.4 packets, an alternative transceiver is required. The Microchip MRF24J40 [153] is a commercial IEEE 802.15.4 transceiver that supports a special operation mode named *turbo mode*, which is compliant to the regulations for radio emissions in the 2.4 GHz ISM band. In this mode, the transceiver operates with a data rate of 625 kbps, i.e., 2.5 times the standard IEEE 802.15.4 rate. This enables the use of these transceivers to synthesize interference using low-cost commercial chips, provided that packets sent in the turbo mode will not be decoded (identified as valid IEEE 802.15.4 packets) by any standard IEEE 802.15.4 stations. In the

next subsections, a description of the PNS built upon IEEE 802.15.4 COTS transceivers is presented, focusing on its architecture, performance and limitations. Furthermore, an evaluation of the *bandjacking* effectiveness using a COTS-based PNS is provided together with a discussion of the obtained results.

3.4.1 Architecture

Low-cost was the main driver for developing a COTS-based PNS. By using a special operation mode of a standard IEEE 802.15.4 transceiver, it was possible to build a PNS that is substantially simpler and cheaper than its SDR-based counterpart, while maintaining the blocking abilities shown by the later. The devised architecture is compact both in terms of hardware and software. This arises from the reduced number of active components and states of the PNS design, as it will be demonstrated in the following subsections.

Hardware

The hardware architecture of the COTS-based PNS was devised around two key components: MCU and 802.15.4 transceiver. As depicted in Figure 3.10, the MCU is the central element of the hardware architecture, bridging the external control and the IEEE 802.15.4 transceivers. The external control block provides input ports tied to MCU internal interrupts, allowing to drive the PNS's state machine by means of external digital signals. The MCU is a Microchip PIC32MX795F512L [154] holding 512 and 128 KB of program memory and RAM, respectively. Besides encompassing large memories, allowing the development of complex applications, this device features a processing core that supports fast context switching and interrupt response, both desirable characteristics for a device that will be employed in a time sensitive application. These features are enabled by its capability of operating with frequencies of up to 80 MHz/105 DMIPS.

The PIC32MX795F512L MCU encompasses several communication interfaces, namely: USB 2.0-compliant full-speed; 10/100 Mbps Ethernet MAC; CAN module 2.0B with DeviceNet addressing support; six UART modules (rate up to 20 Mbps); five I^2C modules (rate up to 1 Mbaud) with SMBus support and up to four 4-wire SPI modules (rate up to 25 Mbps). These SPI features allow the MCU to be easily interfaced to the three IEEE 802.15.4 transceivers required by the PNS. The use of independent SPI ports not only allows a more straightforward control of the IEEE 802.15.4 transceivers, but also enables driving the transceivers simultaneously, e.g., to load the transmission buffers with data to

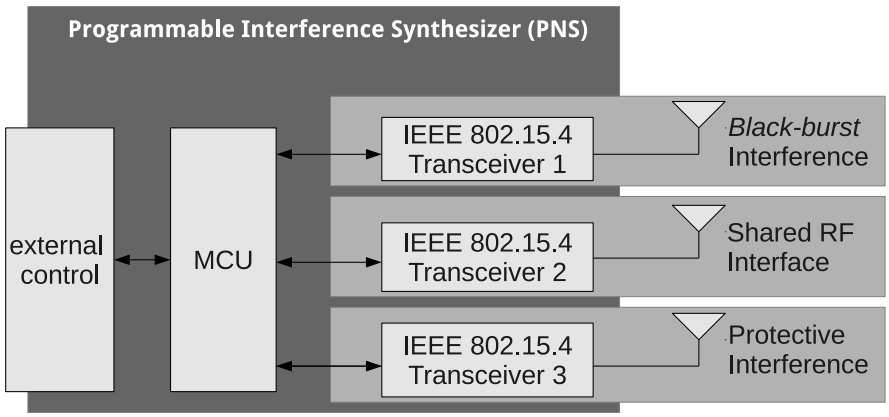


Figure 3.10: COTS-based PNS hardware architecture

transmit. Furthermore, provided that the PIC32MX795F512L MCU integrates five 16-bit digital timers, it is a suitable option for applications requiring a fine timing control such as the development of a PNS.

As indicated in Figure 3.10, three IEEE 802.15.4 transceivers were used in the design of the PNS. The use of the MRF24J40MC transceiver modules [155] is justified by their ability to support the *turbo mode*, where data packets are transmitted at a rate of 625 Kbps, thus preventing them from being perceived as valid by neighboring IEEE 802.15.4 stations. The MRF24J40MC module builds on the features provided by the MRFJ40 transceiver [153] and adds a -108 dBm typical sensitivity (-23 dBm maximum input level) together with a $+19$ dBm typical output power (45 dB transmission power control range). The MRFJ40 is a small sized, low-power, low-cost IEEE 802.15.4 compliant RF transceiver that operates in the 2.4 GHz ISM band and supports the development of applications based on ZigBee [156], MiWi [157] or proprietary protocols. It can be driven by a simple four-wire SPI interface and has hardware support for the CSMA/CA, automatic acknowledgement and packet retransmission mechanisms, along with integrating a hardware security engine (AES-128).

The duration of the longest IEEE 802.15.4 packet (133 bytes) transmitted in the *turbo mode* is 1702 μs, which is represented in Figure 3.11 as $t_{PKTMaxLen}$. Furthermore, this Figure documents a scheme that can keep the medium occupied with packets in two different channels (*black-burst* and protective) simultaneously, using only three transceivers. In this sense, two MRF24J40MC transceivers are used exclusively to perform long packet transmissions (P_L and B_L) on either the *black-burst* interference channel (MRF1) or protective interference channel (MRF3), while the remaining transceiver (MRF2) transmits shorter packets (P_S and B_S) alternately in the *black-burst* and protective interference channels to

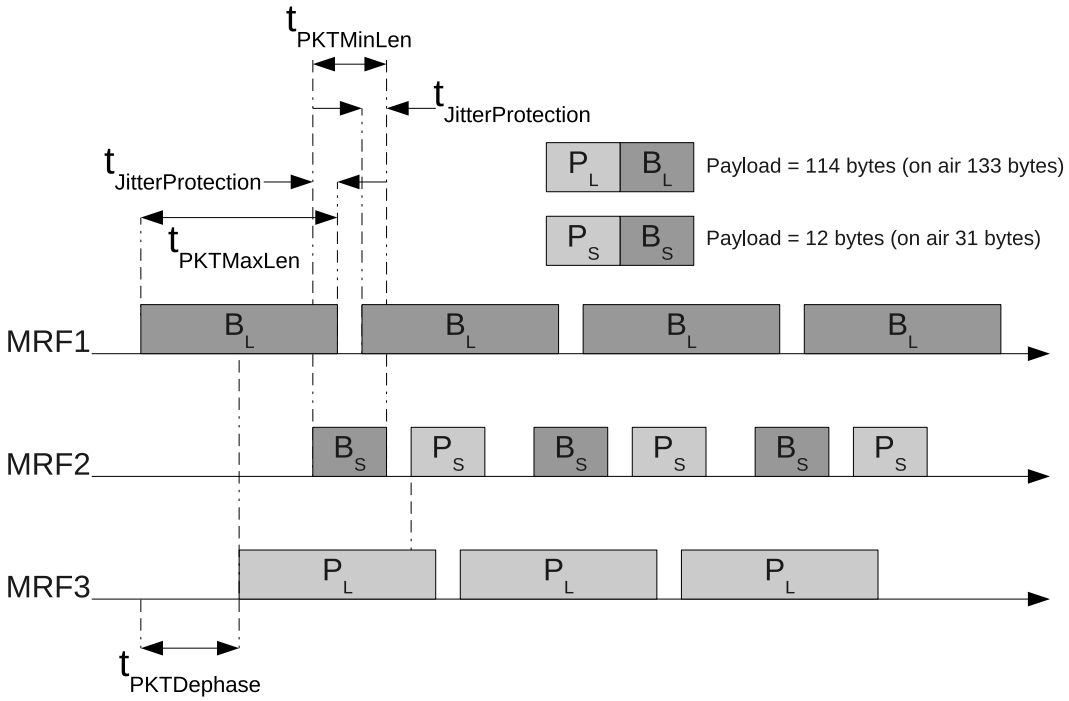


Figure 3.11: COTS-based PNS transceiver operation

cover the idle periods between long packet transmissions, in a round-robin fashion. These shorter packets have a duration of 396.8 μs and are represented by $t_{PKTMinLen}$ in Figure 3.11.

The transceiver operation encompasses an initial dephase of 900 μs that is represented in Figure 3.11 as $t_{PKTDephase}$. This delay allows to synchronize the long packet transmissions on both *black-burst* (B_L) and protective (P_L) channels to occur approximately in the middle of each other. Short packet transmissions are performed by the transceiver MRF2 between consecutive long transmissions to keep the medium busy with interference in both channels. The length of these transmissions was selected so that it permitted a jitter protection interval ($t_{JitterProtection}$) at the beginning and end of the long packets, as documented in Figure 3.11.

The design of this PNS, encompassing three IEEE transceivers operating in the *turbo mode*, was conducted under some assumptions, namely:

- A MRF24J40MC transceiver operating in the *turbo mode* shows negligible latency to initiate a transmission after the trigger command is issued for a packet already loaded to the transceiver's transmission buffer;

- The duration of a *turbo mode* transmission is bounded not only by the trigger command event that initiates it but also by the transceiver's interrupt that signals the end of the transmission;

- The jitter of executing a SPI command at the MRF24J40MC transceiver is negligible when compared to the latency of its transmission.

The MRF24J40MC module allows a broad configuration range using SPI commands. One element that can be controlled via a SPI command is the permanent setting of the MRF24J40 transceiver to transmit mode. With this setting it is possible to shorten the time elapsing from a packet transmission trigger command and the instant when the packet is effectively broadcasted in the medium, because there is no delay associated to the transceiver switching from the receive (default) to the transmit mode. The state of the MRF24J40MC module's power amplifier is another key element that can be controlled via a SPI command. It renders possible to enable/disable the medium broadcast of any packet being transmitted by the MRF24J40 transceiver. Hence, in the context of the developed PNS, this feature is used to start/stop the transmission of interference in both *black-burst* and protective channels. In other words, the PNS application is capable of managing the on/off state of the three MRF24J40MC module power amplifiers, allowing to forward the packet transmissions to the medium or not. In the following subsection, the software architecture of the devised PNS is presented.

Software

The software architecture of the PNS is very straightforward. The application is directly built on top of the hardware interface layer, which encompasses the drivers used to control the layer below, the hardware. The PNS embedded application was developed in the C language, with the libraries provided by Microchip, and compiled using the Microchip's MPLAB C32 compiler. The application begins by configuring the MCU digital ports (direction and interrupts) and peripherals (timers and SPI ports), before initializing the three MRF24J40MC transceivers to operate in the *turbo mode*.

A brief explanation of the general operation of the PNS is presented before discussing the associated state machine in detail. As documented in Figure 3.11, the *turbo mode* transmissions in both *black-burst* and protective channels are performed in a almost continuous fashion. However, they are only broadcasted to the medium when the MRF24J40MC power amplifiers are enabled. Otherwise, these packet transmissions do not reach the

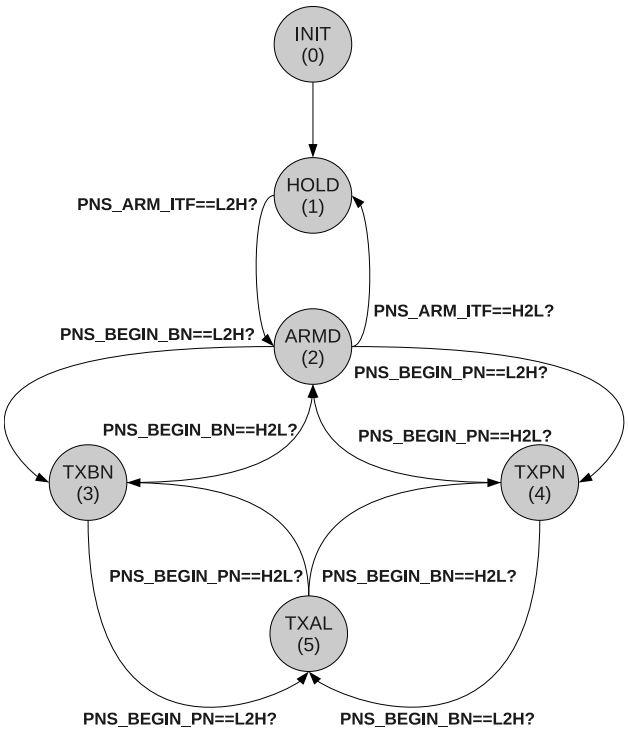


Figure 3.12: COTS-based PNS state machine

antenna and, therefore, are not propagated into the medium. Provided that the power amplifier integrated in the MRF24J40MC modules has a short response time, the PNS directly controls the power amplifier's enabled state in order to start/stop the broadcast of interference in a timely manner. In other words, the PNS schedules quasi continuous packet transmissions in the two predefined channels, with its propagation to the medium being conditioned by the state of the MRF24J40MC power amplifiers', driven directly by the PNS.

The operation of the PNS is characterized by the state machine shown in Figure 3.12. The transitions between states are driven by external signals, which are tied to the MCU's interrupt pins. Both transitions low-to-high (L2H) and high-to-low (H2L) on these pins trigger the occurrence of interrupts and, consequently, of changes in the state of the PNS. As documented in Figure 3.12, the PNS can operate in six states, namely:

INIT: Initialization state. In this state, the MCU hardware and peripherals are initialized along with the transceivers;

HOLD: Hold state. The PNS is idle in this state. Hence, no packets are generated by the transceivers or propagated by the associated power amplifiers;

ARMD: Armed state. The PNS synthesizes a continuous flow of transmissions in both *black-burst* and protective channels. However, such packets are not broadcasted into the medium because all the transceivers' power amplifiers are disabled;

TXBN: Transmit *black-burst* interference state. In this state, the PNS generates a continuous stream of packets aimed at both the *black-burst* and protective channels, but only the power amplifiers associated to the transmissions of *black-burst* interference are enabled. Hence, only this type of interference is propagated into the medium;

TXPN: Transmit protective interference state. A continuous stream of packets aimed at both the *black-burst* and protective channels is generated in this state. However, only the power amplifier of the transceivers that transmit in the protective channel is enabled, thus allowing the propagation of this type of interference;

TXAL: Transmit both *black-burst* and protective interference state. In this state, a continuous packet flow in both *black-burst* and protective channels is broadcasted, provided that all the transceivers' power amplifiers are enabled.

As depicted in Figure 3.12, three signals drive the PNS state machine: PNS_ARM_ITF, PNS_BEGIN_BN and PNS_BEGIN_PN. The first enables (low-to-high transition) or disables (high-to-low transition) the generation of *turbo mode* packets in the three transceiver modules. The two latter signals control the propagation of these packets to the corresponding antennas in the *black-burst* and protective channels, respectively. The state machine progresses in response to the stimuli of the external signals. Hence, after the initialization is concluded, it switches to the HOLD state, where it remains until a low-to-high transition occurs in the PNS_ARM_ITF signal. Thereupon, the state machine progresses to the ARMD state. In this state, it can go back to the HOLD state if a high-to-low transition occurs in the PNS_ARM_ITF signal; to the TXBN or to the TXPN state if a low-to-high transition is forced in the PNS_BEGIN_BN or PNS_BEGIN_PN signals, respectively. Figure 3.12 documents that high-to-low transitions in these signals lead the progress of the state machine in the opposite direction. The state machine progresses to the TXAL state when both types of interference are activated with low-to-high transitions in both PNS_BEGIN_BN and PNS_BEGIN_PN signals.

After implementing this state machine on the PNS's MCU, a few informal tests were conducted to assess its effectiveness. These tests were performed with a PNS attached to

a host standard station, which was responsible for controlling the state machine's driving signals. The test simply consisted on transmitting a *black-burst* interference sequence (by the PNS) immediately followed by an IEEE 802.15.4 data packet transmission (by the standard station) and checking if the data packet was correctly received by a neighboring IEEE 802.15.4 station. The results of these tests revealed that a significant amount of packets were lost. In order to isolate the problem, the PNS was turned off and the tests were repeated. The conclusion was that all the transmitted packets were received and, therefore, the cause of the errors in the former trials were the PNS transmissions, which seemed to overlap the data transmissions. In consequence, we performed an experimental evaluation of the assumptions used for the design of the PNS hardware. In the following subsection, a characterization of the response limitations of the Microchip's MRF24J40 transceiver and of the PNS is presented.

3.4.2 Evaluation

The analysis of both the MRF24J40 transceiver and PNS response limitations was performed using a MRF24J40 based device customized to monitor the medium energy in one or more IEEE 802.15.4 channels of the 2.4 GHz ISM band. This device was named BeeMon due to its ability to monitor ZigBee transmission channels. The details of its architecture and operation can be consulted in Appendix D.

Transceiver Response Limitations

This subsection presents the methodology used to assess the MRF24J40 transceiver timeliness together with the collected experimental results. The results are analyzed and justified accordingly.

Methodology

The characterization of the transceiver response in the *turbo mode* was performed using signals showing the duration of a long packet transmitted by the PNS in two different perspectives. The first signal represents the perspective of the MRF24J40 host controller MCU and it is obtained using a PNS MCU digital output that is set to the logic state "1" when the *turbo mode* packet transmission is triggered (after the packet being loaded into the transceiver's transmission buffer) and reset to the logic state "0" when the associated interrupt is detected, marking the end of the transmission. This signal is connected to

the channel 1 of an HP 54602B 150 MHz digital storage oscilloscope. The second signal is obtained from the output of the BeeMon's low-pass filter and it is connected to the oscilloscope's channel 2.

The oscilloscope was used to register several delay parameters of the transceiver and also to have an informal measure of their jitter by visual inspection. The adopted approach to register the parameters and their jitter follows. The PNS was programmed to perform a long packet transmission (133 bytes overall) in the *turbo mode* every ten seconds. During a period of 100 seconds, for each packet transmission, the oscilloscope synchronizes to the digital output of the PNS and also shows the output of the BeeMon monitor. During this period, the oscilloscope is adjusted to show the intended delay and a photography is taken in each synchronization. At the end of the test procedure, all photographies are discarded, except for the one registering the shortest delay. The value of the longest delay is also annotated, providing an estimate of the maximum delay variation (jitter). An individual testing procedure encompassing the measurement of the delay associated to the transmission of 10 packets was conducted for each of the four parameters under evaluation: transmission trigger and effective propagation delay; transmission interrupt and effective end delay; effective transmission duration and MCU perception of the transmission duration.

Results

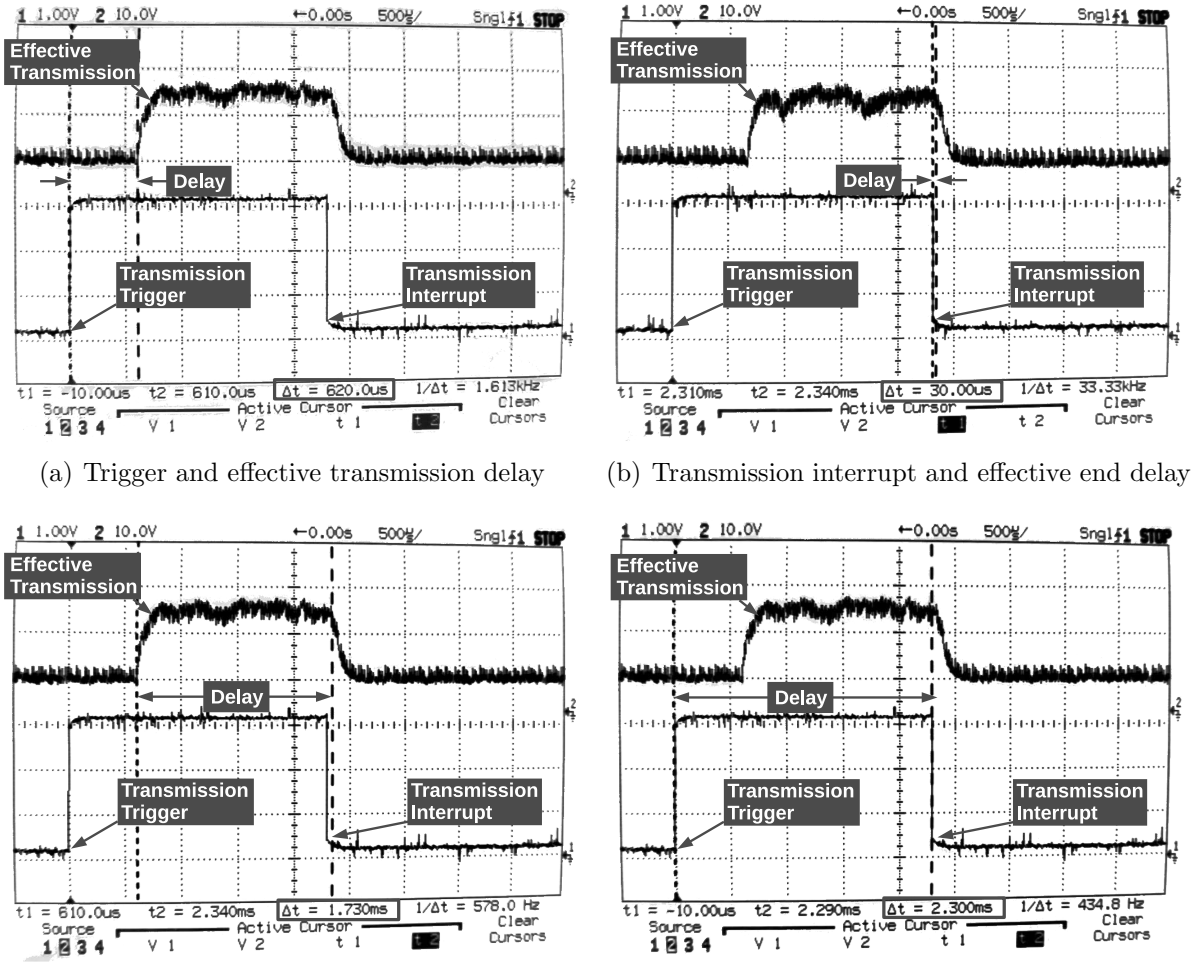
Figure 3.13 presents four captures with overlays identifying signals and instants of interest of the four considered timing parameters. The first timing parameter to be evaluated was the time that elapses between the SPI command instructing the transceiver to perform the transmission of a preloaded packet and the beginning of the effective transmission. As documented in Figure 3.13(a), this delay has a value of 620 microseconds. The observed worst-case jitter was of 52 microseconds (not depicted). The second parameter to be evaluated was the delay between the transmission interrupt and the effective end of the ongoing transmission. As shown in Figure 3.13(b), the effective transmission ends 30 microseconds after the interrupt. The observed worst-case jitter was of 70 microseconds (not depicted). The effective duration of the long packet transmission was the third timing parameter being evaluated. Figure 3.13(c) reports an effective duration of 1730 microseconds with a jitter of 28 microseconds (not depicted). Finally, the last MRF24J40 transceiver parameter to be evaluated was the duration of a long packet transmission, as perceived by the transceiver's host controller, in this case the PNS MCU. This duration has been measured

Table 3.4: MRFJ40 transceiver response timings in the *turbo mode* (corrected)

Parameter	Delay (μs)	Jitter (μs)	Worst-case delay (μs)
Trigger and effective transmission	555	36	591
Transmission interrupt and its effective end	−35	54	19
Effective transmission duration	1730	28	1758
MCU perception of the transmission duration	2300	83	2383

to be of 2300 microseconds (shown in Figure 3.13(d)) with a jitter of 83 microseconds (not depicted). The first two parameters (Figures 3.13(a) and 3.13(a)) were measured using two different signals. However, one of them (BeeMon device) introduces a constant latency of 65 microseconds and a maximum jitter of 16 microseconds, as discussed in Appendix D. With this information and assuming that both the BeeMon delay and jitter are additive, the timing parameters can be corrected accordingly, resulting in the values represented in Table 3.4.

The maximum value of the delay occurring between the packet transmission trigger and the effective beginning of the transmission is very significant (591 microseconds). It renders the PNS implementation unfeasible to synthesize simultaneous interference in two channels using only three transceivers. Provided that this "start transmission" delay is similar for both long and short packets, the MRF2 transceiver, responsible for alternately filling the gaps in the two interference channels, as represented in Figure 3.11, will not be able to generate packet transmissions faster enough to cover the idle periods simultaneously in both channels. Since the "start transmission" delay can reach a maximum of 591 microseconds, the length of the MRF2 transmission must have, at least, the same value. From Figure 3.11 it is possible to conclude that, during an interval of 2940 microseconds comprehending a "start transmission" delay on the MRF1 transceiver (591 microseconds), the corresponding effective transmission (1758 microseconds) and another "start transmission" delay, the MRF2 transceiver must be able to perform three effective packet transmissions (two in the same channel of the MRF1 and one in the MRF3's channel). Considering a "start transmission" delay of 591 microseconds and a (best-case) effective duration of the same value, this procedure will last for 2955 microseconds (5×591 microseconds). Hence, using the proposed approach, it is not possible to guarantee a continuous occupation of two different channels simultaneously. To achieve this ability, the PNS would have to employ four transceivers and use them in pairs to alternately perform transmissions on the selected



(a) Trigger and effective transmission delay

(b) Transmission interrupt and effective end delay

(c) Effective transmission duration

(d) MCU perception of the transmission duration

Figure 3.13: Transmission timeliness of the MRF24J40MC transceiver in *turbo mode*

channel.

The delay between the transmission interrupt and the effective end of the ongoing transmission should be negative, meaning that the interrupt occurs after the transmission effective end, or zero. However, a worst-case value of 19 microseconds was observed during the trials as a consequence of the significant jitter affecting this parameter. This is problematic because it means that a transmission may extend over its designated timing window, affecting other transmissions that may have been initiated.

The effective duration of the long packet transmission can be as high as 1758 microseconds. Although slightly larger than the theoretical duration of 1702 microseconds, provided the measurement scale of 500 microseconds, the deviation seems acceptable, as

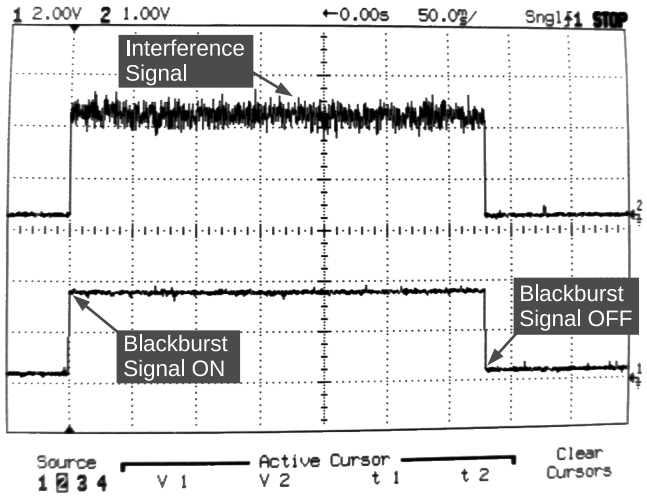


Figure 3.14: Noise sequence transmitted by the COTS-based PNS

it can be a consequence of an oscilloscope adjustment error. The duration of the packet transmission, as perceived by the transceiver's host controller, of 2383 microseconds, in the worst-case, also seems to be in line with its theoretical value, which corresponds to the sum of the delay occurring between the trigger of the packet transmission and its effective energy propagation in the medium (591 microseconds), and the worst-case duration of the effective transmission (1758 microseconds).

PNS Response Limitations

In this subsection, the methodology used to assess the PNS timeliness is presented as well as the experimental results obtained using the devised testbed. As before, these results are discussed and justified accordingly.

Methodology

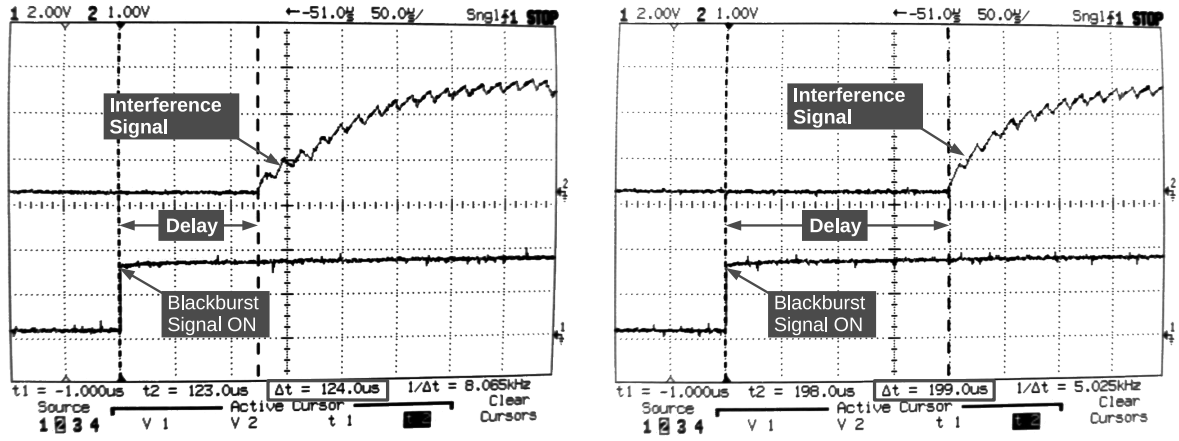
The PNS response characterization was conducted similarly to the MRF24J40 transceiver. Figure 3.14 illustrates an interference signal generated by the PNS and measured using the BeeMon monitor. The goal of this evaluation is the characterization of the PNS turn on and turn off delays. For this purpose, the PNS was programmed to perform a periodic *black-burst* interference transmission with a fixed duration of 330 milliseconds (see Figure 3.14) and a period of 10 seconds. The external signal driving the PNS on/off state was connected to the channel 1 of a HP 54602B oscilloscope, while the signal coming from the BeeMon filter output was tied to the oscilloscope's channel 2. The testing procedure was

conducted over an interval of 100 seconds and, for each interference period, the oscilloscope is synchronized to the driving input of the PNS. During the testing procedure, a photography is taken in each synchronization showing a given delay. After the conclusion of the test procedure, the photographies having the shortest and longest delays are kept. Using this methodology, two PNS delay parameters were evaluated: the interference start and stop latencies.

Results

Figure 3.15 depicts the minimum and maximum PNS interference start delay. As shown, the minimum and maximum delays are of 124 and 199 microseconds, respectively. Although the measurement of these delays can be affected by the BeeMon's maximum latency of 81 microseconds, the start delay range is nonetheless high, with a significant delay variation. This can be justified by the actual implementation of the PNS. Currently, to begin generating interference, the PNS must be at the ARMD state and either one of the PNS_BEGIN_PN or PNS_BEGIN_BN, must be driven high. This occurrence triggers an interrupt on the PNS MCU, whose ISR is programmed to send a command to the associated transceiver(s) (via SPI) to make it (them) turn the embedded power amplifier on, thus propagating the interference signal being generated into the medium. This approach was taken mainly because it renders possible building the PNS using standard unmodified IEEE 802.15.4 modules, thus avoiding a demanding RF certification procedure that, apart from having a high cost, comprehends the risk of being denied. The compliance with the RF regulations is a critical aspect of any commercial application employing wireless communications. Hence, by using pre-certified modules without any tampering, a specific certification procedure could be avoided while maintaining the PNS RF compliance by inheriting the module's characteristics already certified.

This approach was also adopted due to the assumption that the state machine of the MRF24J40MC transceiver would be fast (and predictable) to respond to commands sent via SPI, in particular the command to enable/disable the power amplifier. However, as these measurements seem to indicate, this assumption is not completely sustained by the results. First, considering that the PNS exhibits a negligible interrupt latency for the external driving signals and a latency of 1.6 microseconds for the transmission of the the "enable power amplifier" via SPI, the minimum delay of 43 microseconds, taking off the BeeMon maximum response time of 81 microseconds, suggests that the transceiver's state machine takes a significant amount of time to process and execute the received command.



(a) Minimum

(b) Maximum

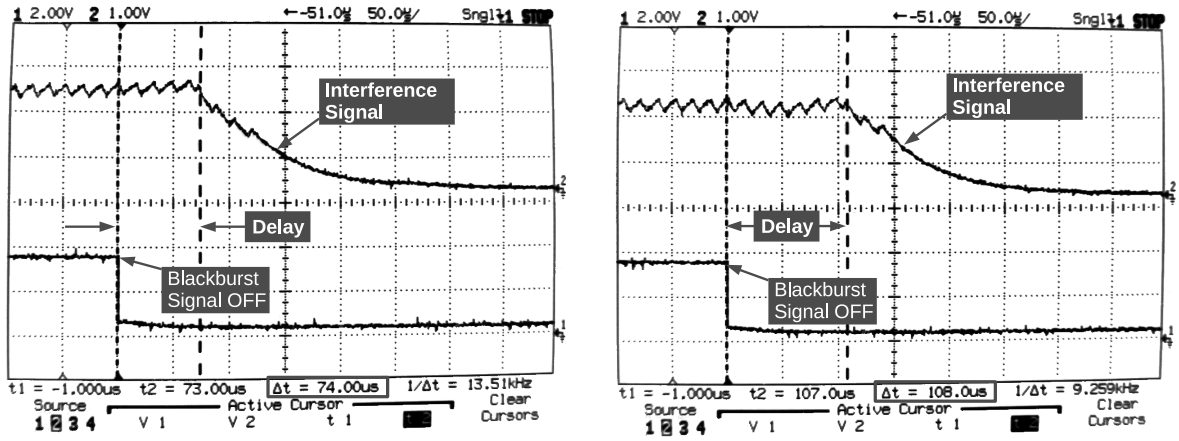
Figure 3.15: COTS-based PNS timeliness: interference start latency

Second, the broad variation of the start delay seems to indicate that the internal state in which the transceiver receives the command has a significant impact on the time it takes to start its processing.

The minimum and maximum PNS interference stop latency is documented in Figure 3.16. As shown, the minimum and maximum delays taken by the PNS to stop interference are 74 and 106 microseconds, respectively. These values are far smaller than those of turning the interference on. Provided that both actions (start or stop) are handled in the same way by the PNS, i.e., a SPI command is sent to the transceiver after the PNS_BEGIN_PN or PNS_BEGIN_BN signal has been driven low, the different response time is caused by the transceiver responsiveness to the "disable power amplifier" command.

Although Figures 3.16(a) and 3.16(b) show the interference signal stopping after the interrupt that marks the end of the transmission, this actually only occurs when the stop delay is larger than the worst-case response time of the BeeMon monitor. This occurs because the BeeMon monitor signal is affected by a delay that can reach the maximum value of 81 microseconds, thus delaying it by this amount of time regarding the external signal driving the PNS. As occurred in the MRF24J40 timing response subsection, the interference generated by the PNS can extend enough to overlap a subsequent transmission, thus causing its corruption. This seems to justify the packet errors encountered in the informal evaluation described in Subsection 3.4.1.

The above mentioned timing limitations of the PNS can be overcome by driving the power amplifier of the transceiver module directly, i.e., by changing the PNS implemen-



(a) Minimum

(b) Maximum

Figure 3.16: COTS-based PNS timeliness: interference stop latency

tation so that, instead of sending a SPI command to turn on/off the power amplifier, a digital signal connected to its enable input is used alternately. The time response of the PNS implementation and of the power amplifier become similar in this approach, which has the disadvantage of requiring the PNS to be submitted to a certification process to consent its commercialization. In the following section, besides the COTS-based PNS implementation evaluation, an assessment of the *bandjacking* technique is presented using a time slack between the end of the interference transmission and the beginning of the critical data packet to avoid its corruption.

3.5 An Evaluation of the Bandjacking Effectiveness

This section aims at assessing the *bandjacking* technique timeliness and effectiveness in supporting critical data communications by using a COTS-based PNS implementation. In this sense, a testbed was devised so as to evaluate the *bandjacking* ability to support real-time communications in uncontrolled environments, where different communication stations may contend for the medium. Provided that the critical data communications are IEEE 802.15.4 based, the technologies chosen to contend for the medium are the IEEE 802.15.4 and the IEEE 802.11.

As defined in Section 3.4.2 (Transceiver Response Limitations), the implementation of the PNS using three transceivers poses the limitation of being only capable of synthesizing one type of interference at a time: *black-burst* or protective. In this sense, a PNS

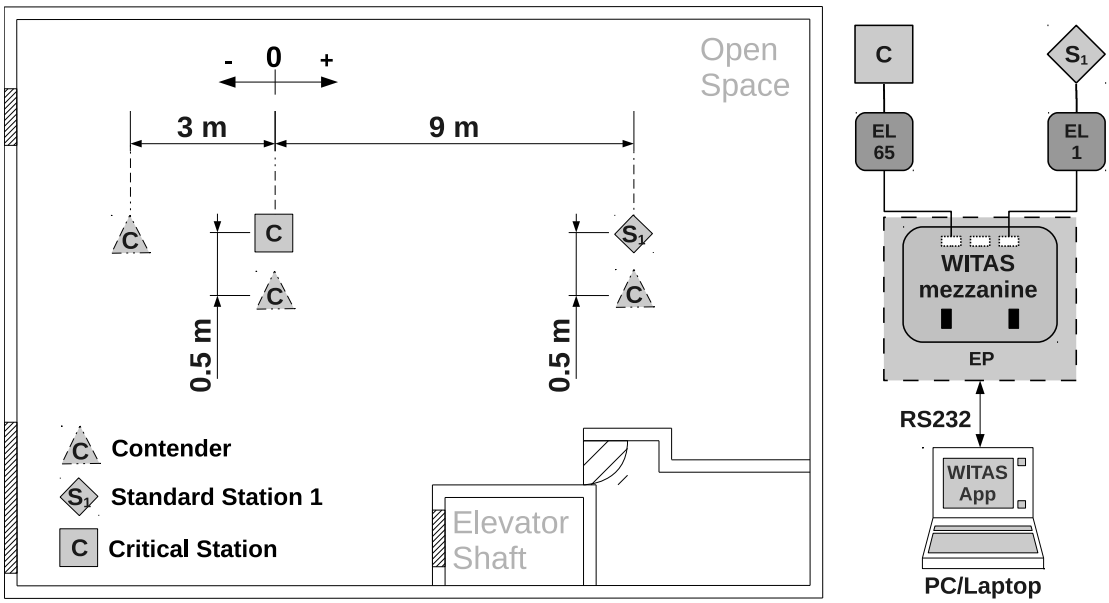
implementation was devised to allow selecting one of the two types of interference to be synthesized. No significant modifications were made to the state machine presented in Figure 3.12, except for the fact that, when a given type of interference is configured, only the associated triggering signal (PNS_BEGIN_PN or PNS_BEGIN_BN) will be sensible to a level change and allow the control of the synthesized interference.

An individual evaluation of the *bandjacking* effectiveness using either *black-burst* or protective interference is presented in this section. Despite the separate use of these types of interference, the conditions in which they are used are similar. This allows to grasp the level of *bandjacking* effectiveness achievable using a PNS that can synthesize simultaneously both types of interference. In the following subsections, a description of the methodology used in the *bandjacking* assessment is presented together with the associated results and their discussion.

3.5.1 Methodology

The methodology employed to assess the *bandjacking* technique was conceived to provide a solid evaluation of its effectiveness. Figure 3.17 depicts the elements that compose the devised testbed. Figure 3.17(a) documents the positions of the devices on the test environment, while Figure 3.17(b) depicts the logical arrangement of the WITAS tool used to evaluate the *bandjacking* effectiveness. As documented, the testbed includes one critical station, one standard station; and one contender station. The standard station is placed at a distance of 9 meters away from the critical station. This is a value close to the standard nominal range (10 meters) of the IEEE 802.15.4 technology. The contender is placed in three different positions: 0.5 meters from the standard station, 0.5 and 3 meters from the critical station. Positioning the contender on these locations renders possible an evaluation of the impact of having the interference source very close to the standard station and to the critical station, or at a reasonable distance from the latter.

The trials were conducted on a sub-basement with an area of approximately 116 m^2 (13.5 m $\times$ 8.6 m). Several photos of the physical space are shown in Figure 3.18. This space benefits from its seclusion due to the thick concrete walls that block interference from the outside of the building. The WITAS tool allows the measurement of several parameters including the packet error rate, which is the focus of this evaluation. It encompasses three types of devices: *Event Loggers*, *Event Processor* and an application running on a PC. The *Event Loggers* are attached to the communication stations of interest and register the packet transmission/reception events that have occurred. The *Event Processor* is directly



(a) Physical arrangement (b) WITAS setup

Figure 3.17: Bandjacking evaluation testbed

driven by the PC application to individually collect the information saved at the *Event Loggers* and forwards it to the application for backup. The WITAS application running in the PC, besides saving the data into a file for later analysis, allows processing the collected data to obtain meaningful statistical information about the communication reliability and timeliness. Appendix A provides detailed information regarding the WITAS architecture, operation and application.

A trial using the WITAS tool encompasses three phases: configuration, execution and data processing. The first represents the interval in which all the elements participating in the trial (communication stations and WITAS devices) are set up to operate during that trial. In this phase, the WITAS application running in the PC assigns to each element a new configuration that, on the PC application side, can be stored on a XML data file for later editing and backup. The automation of the trials allows performing the configuration process in a timely fashion. The execution of the trial is driven by WITAS PC application using specific commands to the elements of the testbed. In this case, the critical station can be configured to start/stop sending critical data packets and to use different transmission periods. The standard station only supports commands for starting/stopping its operation. During a trial, both communication stations log all the transmission and reception packet events into the associated *Event Loggers*, which forward this information to the *Event*

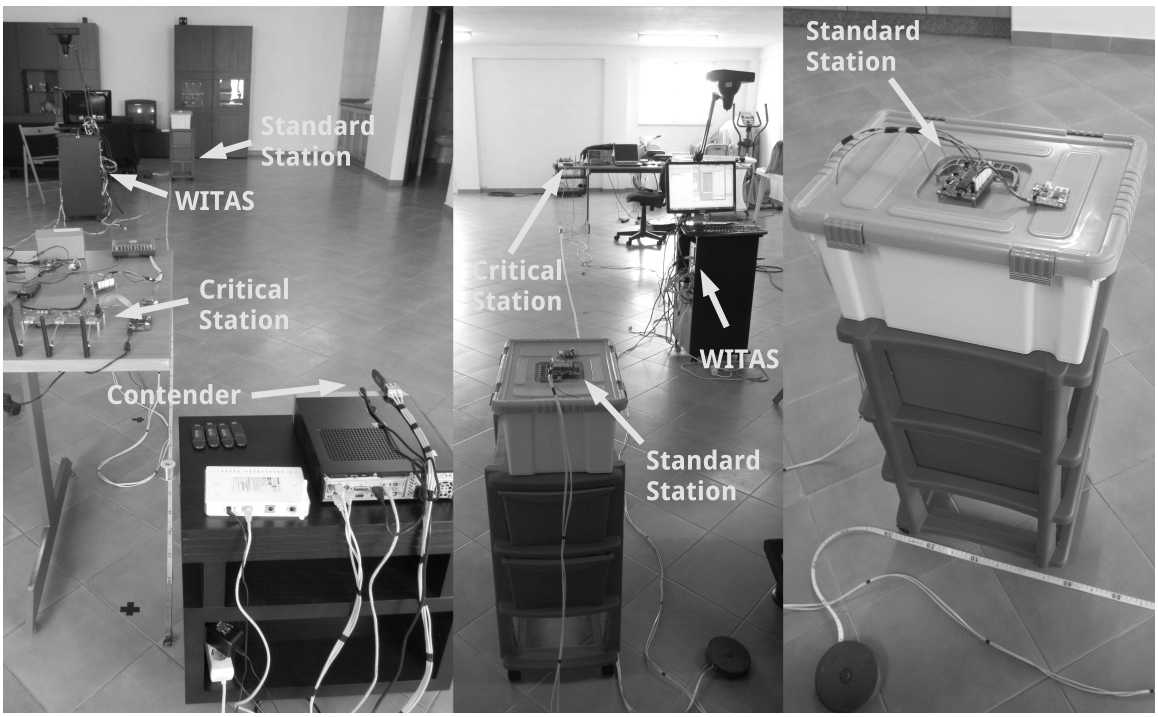


Figure 3.18: COTS-based trials

Processor and then to the application running in the PC. At the end of the trial, the data temporarily stored in the WITAS application (memory) can be permanently saved into a file for later analysis. In the last phase, the WITAS PC application uses the collected data to calculate the packet error rate experienced in the trial.

Figure 3.17(b) shows the connection of the *bandjacking* communication elements to the WITAS evaluation tool. As documented, the standard station S_1 and the critical station are connected to the Event Logger EL 1 and EL 65, respectively. These *Event Loggers* are then linked to the Event Processor (EP), which, in turn, is tied to the PC using a serial port. This connection supports a bi-directional interface between the WITAS application running in the PC and the *Event Processor*. In this sense, the *Event Loggers* record locally the events (packet transmission or reception) reported by the associated communication stations and send them periodically to the *Event Processor*. The transmission of these events is triggered by specific command messages sent from the *Event Processor* to the *Event Loggers* in a round robin fashion.

The *bandjacking* implementation was configured to provide a maximum slack interval of 200 microseconds between the end of the interference sequence produced by the PNS and the beginning of the critical transmission performed by commercial transceiver in order to

avoid the overlapping of these transmissions. This value was experimentally verified to be a good compromise between providing enough room to cope with the PNS and MRF24J40 transceiver limitations; and the requirement of shortening the capture interval to a minimum. Furthermore, the testbed was configured to conduct trials encompassing 1000 packet transmissions with a 1 second period between them. The *bandjacking* effectiveness assessment was carried out using the general parameters shown in Table 3.5 and the testbed elements positioned according to Figure 3.17(a). The contender was placed on a mobile MDF stand with a height of 64 centimeters, the critical station on a MDF table with a height of 82 centimeters and the standard station on a mobile plastic stand with a height of 112 centimeters.

Provided that this evaluation aims at assessing the effectiveness of the *bandjacking* technique to support real-time communications in uncontrolled environments, two types of interference were selected to contend for the medium: IEEE 802.15.4 and Wi-Fi (IEEE 802.11). The contenders were configured to perform transmissions in the Wi-Fi channel 1 (CAOS) and in the IEEE 802.15.4 channel 14 (ZigFlooder). The critical transmissions are also performed in channel 14, thus overlapping both types of contender transmissions. The CAOS contender used in these trials employs a Wi-Fi dongle configured to perform transmissions in channel 1 with a power of 20 dBm and a period of 1 millisecond. The ZigFlooder uses a uMRFs board programmed to perform contention transmissions in channel 14 with a power of 0 dBm and a period of 10 milliseconds. These contention transmissions employ the "Energy Above Threshold" CCA mechanism with limit of -69 dBm. For an in-depth analysis of the CAOS, please consult Appendix B. Both critical and standard stations were configured to perform their data transmissions in channel 14 with two different power levels: 0 dBm, −10 dBm. The use of these power levels was motivated by the need to assess the impact of the transmission power, which is coupled with the distance between transmitter and receiver, in the effectiveness of the *bandjacking* technique.

Table 3.5: Bandjacking evaluation: general parameters

Parameter	Unit/Type	Contender		Critical Station		Standard Station
		CAOS	ZigFlooder	Data	PNS	
Height	cm	64		82		112
Channel	IEEE 802.15.4	—	14	14	11, 14	14
	IEEE 802.11	1	—	—	—	—
Power	dBm	20	0	0, -10	18	0, -10

As introduced, the PNS is only able to produce one of the two types of interference at a time: *black-burst* or protective. In this sense, part of the trials were conducted with the PNS configured to synthesize interference on channel 14 (*black-burst*) while the other part were realized with interference on channel 11 (protective). In both cases, the PNS was configured to send packets with a transmission power of 18 dBm. Performing critical data communications supported on a PNS that can only transmit one type of interference at a time is a limitation in the analysis of the *bandjacking* effectiveness. However, since the same set of trials are performed under similar conditions employing both types of interference, it is possible to estimate the effectiveness of the *bandjacking* technique if a PNS producing simultaneously the two types of interference was used instead.

3.5.2 Results

Table 3.6 shows the Packet Error Rate (PER) of the critical data communications using the setup presented in the previous subsection. As documented, three main scenarios were evaluated concerning the PNS usage: PNS off, PNS generating *black-burst* interference and PNS generating protective interference. In the *black-burst* interference scenario, two types of contender have been evaluated: Wi-Fi and IEEE 802.15.4. The first allows to assess the robustness of the critical data transmissions in an uncontrolled environment affected by Wi-Fi noise on an overlapping band. The second allows evaluating if the *bandjacking* technique is capable of protecting the critical data transmissions from contending stations also employing the IEEE 802.15.4 technology. In the protective interference scenario only the immunity to Wi-Fi noise was evaluated. This occurs because, typically, only wireless contending technologies whose bandwidth covers the protective interference channel will be blocked by its transmissions.

The scenario encompassing both PNS and contender turned off was evaluated to provide a reference measure of how the critical data transmissions perform in an environment free from interference. It also allowed to confirm if the slack interval between the interference sequence and the subsequent data packet was long enough to avoid its corruption due to the PNS jitter. As demonstrated, the PER is of 0.1 % for both power levels (0 and -10 dBm), which indicates that one packet was lost in the trial's 1000 transmitted packets.

As introduced, considering the case where the PNS synthesizes *black-burst* interference, critical data communications are evaluated under either Wi-Fi or IEEE 802.15.4 noise. In the first case, as depicted in Table 3.6, a small PER ($\leq$ 0.3%) occurs for a transmission power of 0 dBm when the CAOS is placed close to the critical station (at 0.5 meters) or at

Table 3.6: Critical station packet error rate (percentage)

PNS	Contender	Power (dBm)	Contender active at @-3m	@CS	@SS	Contender Inactive
Off	Off	0	NA	NA	NA	0.1
		-10	NA	NA	NA	0.1
BB	Wi-Fi	0	0.2	0.3	95.3	NA
		-10	97.1	86.3	72.8	NA
	802.15.4	0	0.1	0.0	0.7	NA
		-10	0.0	0.0	2.5	NA
PI	Wi-Fi	0	0.5	0.2	0.4	NA
		-10	0.3	0.3	1.0	NA

CS *Critical Station* SS *Standard Station*
BB *Black-burst Interference* PI *Protective Interference*

−3 meters from it. However, when the contender is set near the standard station, the PER becomes much higher, around 95.3 %. A similar PER increase occurs when the critical data transmission power is −10 dBm. In this case, the PER is of 97.1 %, 86.3 % and 72.8 %, corresponding to the placement of the CAOS at −3 meters from the critical station, near the critical station or near the standard station, respectively.

The interpretation of these results follows. The small PERs observed when the contender is placed at −3 meters or close to the critical station can be justified by the sporadic occurrence of Wi-Fi transmissions during the capture interval, which is a consequence of the slack interval introduced to avoid having an overlap between the *black-burst* and the subsequent data packet transmission. When the capture interval lasts more than 10 microseconds, there is a chance for the medium to be used by a Wi-Fi station to perform a packet transmission. This bound corresponds to the maximum amount of time that the medium can be in the idle state before risking a Wi-Fi station transmission [158]. However, because the slack interval is relatively small, the observed packet error probability is also small. Regarding the higher PER, it seems to be a consequence of the decrease in the Signal-to-Noise Ratio (SNR) observed at the receiver, which occurs when the contender is placed nearby the standard station, or as an outcome of reducing the critical station transmission power. This result seems to be aligned with the trend observed in BER, as a consequence of the decrease of the SNR in [120].

The results documented in Table 3.6 show that the PER is negligible (0.1 % or less) in the scenarios where the ZigFlooder is placed far from receiver when the critical data

communications are protected with *black-burst* interference. The PER suffers a small increase (to 0.7 %) when the contender is placed close to the receiver and further increased when the critical station's transmission power is reduced to −10 dBm (2.5 % in this case). Provided that an IEEE 802.15.4 transmission can only be initiated if the medium is idle for a period of, at least, 128 microseconds [158], even with a slack of 200 microseconds in the capture interval, the standard IEEE 802.15.4 station (ZigFlooder) will not be able to find the medium idle and initiate any transmissions frequently, possibly due to a hardware response time limitation of the "alien" station. This justifies the negligible PER values found when the contender is far from the receiver station. Regarding the higher PER of 0.7 % and 2.5 %, as before, the PER becomes higher with a decrease in the SNR at the receiver. Since the contender uses a communication technology with a similar transmission power to the critical station, it is expected that it will have a higher impact when placed near the receiver.

Finally, as documented in Table 3.6, when the critical data communications are evaluated under Wi-Fi noise, having the PNS synthesizing the protective interference, results show that the PER is equal or smaller than 1.0 %. In this scenario, the PNS continuously produces interference in a channel different from the one used in the critical data communications. The PNS interference covers a spectrum region common to the selected Wi-Fi channel, thus making the CAOS find the medium always occupied and inhibiting its transmissions. This justifies the small PER observed in the results. Nevertheless, the PER is not negligible. This can be explained by the sensitivity pattern of the Wi-Fi contender, which may have different gains in different reception directions. Hence, due to a gain minimum in the direction of the CAOS, the contender may sporadically evaluate the medium idle, even if a protective interference sequence is being transmitted.

The results presented in Table 3.6 demonstrate that the *bandjacking* technique can be effectively implemented using commercially available low-cost components. These results indicate that the use of *black-burst* interference can significantly contribute to the immunity of IEEE 802.15.4 critical data transmissions in environments susceptible of IEEE 802.15.4 contention. Moreover, it is possible to ensure that Wi-Fi stations have a negligible impact on IEEE 802.15.4-based critical data transmissions by employing protective interference in an adjacent channel. Overall, this section indicates that the support of low-power deterministic communications is possible using the IEEE 802.15.4 technology in open environments, where other technologies may dispute the medium. This can be ensured by using the *bandjacking* technique implemented with a responsive PNS (turn on/off delays smaller

than 10 microseconds) capable of synthesizing (and controlling) simultaneously *black-burst* and protective interference.

3.6 Summary

This chapter presented a review of the literature regarding *black-burst* contention and identified the key proposals with focus on their advantages and limitations. Furthermore, it described the *bandjacking* MAC technique using both formal and informal approaches. The explanation of the *bandjacking* technique encompassed the analysis of a reference architecture as well as of its intended operation. Besides a theoretical reference architecture, this chapter proposed two physical implementations for the key element of the critical station: the PNS. In this sense, one of the described implementations was based on a SDR approach, while the other was based on a COTS approach. The first was used to evaluate to which extent was it possible to devise a pseudo IEEE 802.15.4 based PNS that would effectively be able to hinder Wi-Fi communications. The first conclusion obtained in this approach was that the interference produced by a SDR-based PNS can effectively block Wi-Fi transmissions being performed in the overlapping channel. The second conclusion is that the protective interference generated using pseudo IEEE 802.15.4 transmissions must be separated from the critical data channel by, at least, two channels in order to avoid cross channel interference.

The PNS COTS implementation was built upon the conclusions of the SDR-based development, but was focused on a commercial realization using COTS. In this sense, in order to study their adoption for the critical station, both the response time limitations of the MRFJ40 transceiver and of the PNS were characterized. In the first case, the conclusion was that the transceiver encompasses a significant latency to initiate transmissions in the *turbo mode*, which renders unfeasible the development of a PNS capable of producing two types of interference simultaneously using only three transceivers. Besides this, the transceiver also exhibits a relatively high delay between the interrupt, which indicates the end of the transmission and its effective end, thus making the transmissions extend over the designated window by a period of time that can corrupt the transmissions occurring in this interval (19 microseconds). In the second case, regarding the PNS time response limitations, it was concluded that the start delay of the interference generated by the PNS ranges from 43 to 118 microseconds while the interference stop latency can reach up to 25 microseconds.

The last part of this chapter is concerned with the evaluation of the *bandjacking* effectiveness using the COTS-based PNS. Although the presented PNS implementation lacks the ability to simultaneously synthesize *black-burst* and protective interference, the results collected individually indicate that the original goals can be met using a PNS capable of simultaneously synthesizing and controlling both types of interference. An aspect currently contributing to lower the *bandjacking* technique determinism is the PNS turn on/off jitter. One way to avoid this problem is by directly driving the PNS transceivers' power amplifiers, which have a shorter response time when compared to the use of SPI commands for this purpose. The results presented in this chapter endorse the conclusion that the support of dependable low-power communications is possible in open environments.

"There's no system foolproof enough to defeat a sufficiently great fool."

Edward Teller (1908 - 2003)

4

The Wireless Flexible Time-Triggered Protocol

In the ever evolving technology domain, distributed architectures have found applications in many areas, ranging from home automation to avionics. The dependability, composability, scalability and maintainability provided by these architectures make them particularly interesting to support complex applications, where significant efficiency gains can be obtained through modularization. In the previous chapter it was demonstrated that the use of the *bandjacking* technique allows enforcing dependable critical transmissions in environments affected by contention-based interference. However, the application of this technique in distributed environments encompassing multiple uncoordinated critical stations would be unfeasible, since the critical data transmissions would be impaired by interference from each other. The *bandjacking* forceful nature is geared to an adoption where a critical station is entitled to the exclusive use of the medium for a predefined period of time. This cannot be guaranteed in scenarios encompassing multiple uncoordinated critical stations.

In order to benefit from the dependability provided by the *bandjacking* technique in a communication protocol, a coordination mechanism between stations must be established to allow their interaction without transmission overlapping. Furthermore, because the interference sequence of a bandjacking access takes a significant amount of time, thus reducing the communication's efficiency, the time spent in performing interference transmissions should be reduced to a minimum. In the following sections, a wireless communication protocol inspired on the Flexible Time-Triggered (FTT) paradigm is presented. This protocol builds on the medium capture and maintenance conveyed by the *bandjacking* technique to support dependable, flexible, low-power and real-time communications in

open environments.

4.1 The FTT Paradigm: A Short Introduction

The FTT communication paradigm [159, 160] was devised to address the requirements of distributed architectures and to support a high level of operational flexibility using a time-triggered approach guaranteeing its timeliness and safety. This communication paradigm can be instantiated into several protocols, depending on the adopted networking technology. Three protocols are currently supported: FTT-CAN [159], FTT-Ethernet [161] and FTT-SE [162], which are based on the CAN, Ethernet, and micro-segmented switched Ethernet technologies, respectively. Although these protocols have different implementations and differ in several technology-related aspects, they share a common set of properties emerging from the FTT communication paradigm.

Figure 4.1 documents the FTT system architecture. It comprises one master and several slave nodes in an asymmetric, synchronous architecture where the master manages and coordinates the network communication activities. The system requirements database (SRDB) represents the information repository where all meaningful data pertaining to the protocol is stored. The master holds both the communication requirements and message scheduling information, which enables on-line admission control with support for update of requirements and message scheduling on-the-fly.

The FTT protocols are characterized by encompassing two temporally isolated types of traffic: time-triggered and event-triggered. The former is called synchronous traffic and is explicitly scheduled by the master. The later, despite being restricted to the specific window defined by the master, is autonomously driven by the slaves and is called asynchronous traffic. The temporal isolation between the two types of traffic is enforced in any protocol implementation. The Asynchronous Messaging System (AMS) and the Synchronous Messaging System (SMS) provide the application services required to manage these traffic types. The first offers basic send and receive services, while the latter supports services based on the producer-consumer model.

The FTT paradigm defines that the bus time is divided in consecutive fixed duration slots, called Elementary Cycles (ECs), which start with the master's broadcast of a periodic trigger message (TM) that synchronizes all nodes. Figure 4.2 depicts an elementary cycle example including the trigger message, four asynchronous messages (AM_1, AM_3, AM_5 and AM_6) and five synchronous messages (SM_1, SM_3, SM_4, SM_{10} and SM_{14}). As depicted,

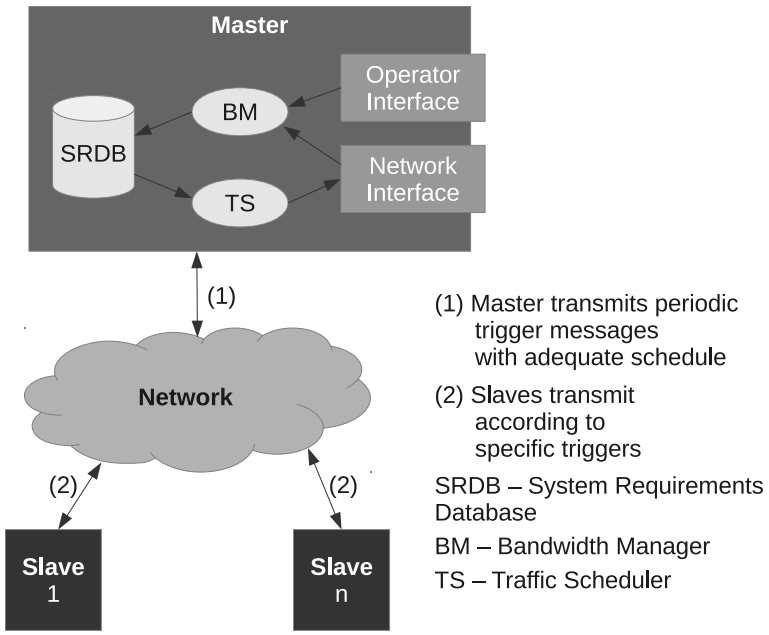


Figure 4.1: Master-slave FTT system architecture

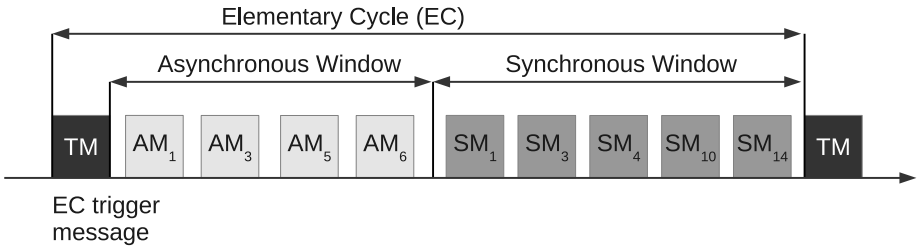


Figure 4.2: FTT elementary cycle

these messages are transmitted in two traffic windows: asynchronous and synchronous. The former is used to convey event-triggered traffic handled according to a best-effort policy. The latter conveys the time-triggered traffic specified in the TM, which is previously subject to admission control. The window order depends on the FTT protocol being employed. For instance, the FTT-CAN protocol sequence is represented in Figure 4.2. Here, the asynchronous window appears before the synchronous, while the other way around occurs in the FTT-Ethernet and FTT-SE protocols, i.e., the asynchronous window appears after the synchronous.

The elementary cycle begins with the broadcast of the trigger message holding the scheduling decisions for the elementary cycle, i.e., the identification of the synchronous messages which must be transmitted in the synchronous window. After decoding the TM, all stations participating in the network verify if they are the producers of any of the

scheduled messages identified in the TM and, if so, they perform the corresponding transmission on the synchronous window of the EC. One key aspect of the paradigm is that the traffic scheduler cannot schedule more messages than those that can fit in the synchronous window. This allows guaranteeing the temporal isolation between elementary cycles. The asynchronous traffic is freely generated by the slave stations in the asynchronous windows using the specific medium access technique adopted for the underlying communication technology. In any case, the transmissions requests that have not been served during the asynchronous window are queued and resubmitted in the asynchronous window of the subsequent EC. More information regarding the FTT paradigm is available in [160].

4.2 Protocol Specification

According to You *et al* [163], the development of a MAC protocol for wireless networks must address a specific set of design objectives, namely: high network throughput; low collision probability (or collision-free) communication; energy preservation; high quality of service; prevention from starvation and fairness within the same priority level; efficient broadcast; and simple hardware requirements. A high network throughput typically contributes to obtain reduced packet delays and, therefore, to a better communication response. Likewise, reducing or avoiding collisions leads to an improved network usage and responsiveness, besides avoiding spending energy with corrupted data transmissions. The energy preservation is particularly stringent in the context of mobile wireless communications, where devices have a limited access to energy supply sources. Because the liveness of such networks is highly dependent on the autonomy of their nodes, the optimization of power consumption allows sustaining their operation for larger periods of time. The quality of service allows differentiating traffic flows according to specific communication parameters such as latency and throughput. In this sense, the support of multiple priority levels enables packets to meet the application's service requirements by allowing those with higher priorities to be transmitted with a shorter delay. The prevention of starvation and fair use of the medium ensures that all stations have the opportunity to perform their transmissions. This is an important design goal because it fosters a balanced use of the shared resource (communication medium) among stations. The efficient broadcast objective is only meaningful in multi-hop wireless networks, as otherwise, it merely depends of the communication range. In such networks, broadcasts are typically used to convey information about the network itself (synchronization and routing) and to propagate data

messages from the neighbors. Hence, effective broadcast mechanisms are required to guarantee that all nodes of the network get the transmitted information. Finally, the simple hardware requirement is of paramount importance when a commercial implementation and deployment is needed. Since the complexity of the hardware is tightly coupled with its cost, the development of a MAC protocol using widely available technologies is fundamental when targeting commercial applications.

Regarding the aforementioned objectives, when demanding requirements are established for a subset of features, a compromise is usually required. In such cases, some features are favored over others, which are adjusted according to the resulting limitations or, in many cases, dropped. The design of the WFTT protocol has taken all the above mentioned objectives into account but has particularly emphasized the group that includes collision avoidance, quality of service, fair use of the medium, energy preservation and low-cost hardware. The following sections identify the key design options.

4.2.1 Architecture

The architecture of a typical WFTT network is represented in Figure 4.3. This example will be used to explain the WFTT protocol in the following subsections. As documented, the network is composed of one critical station (CST1), three standard stations (SST2, SST3 and SST4), and one "alien" station (AST1). "Alien" stations, as previously addressed, are contention-base communication devices that do not belong to the WFTT network. These stations can employ the same technology of the WFTT stations or a different technology. However, they are configured to perform transmissions on a channel overlapping the region of the spectrum being used by the WFTT network. Furthermore, "alien" stations can attempt to initiate a transmission in any random instant of time. Provided the use of a contention scheme to access the medium, the transmission will be conveyed if the medium is perceived idle for a period of time long enough.

The critical station (CST1) encompasses a standard station (SST1) and a programmable interference synthesizer (PNS1). The first is responsible for communicating the critical data and includes a communications controller (CC1) and a commercial transceiver (CT1). The second is engaged in synthesizing two types of interference (*black-burst* and protective) according to pre-defined profiles, as depicted in Figure 3.2. The standard stations (SST1, SST2, SST3, and SST4) are assumed to be capable of performing transmissions in a TDMA fashion by disabling any existing CSMA/CA mechanism. This allows improving the latency and the jitter associated to the standard station transmissions, besides enabling their

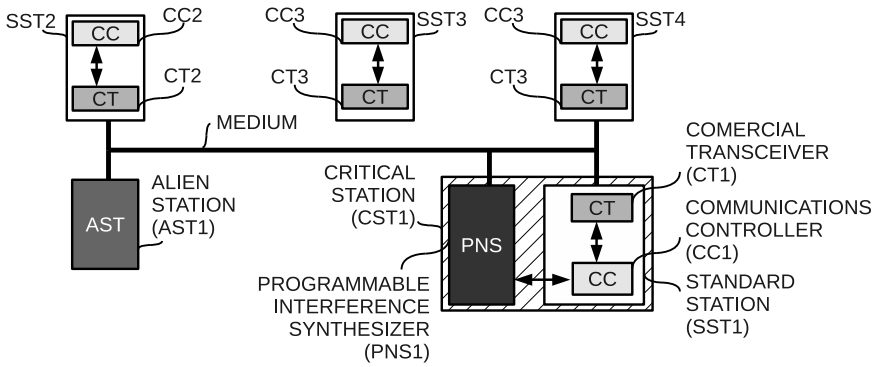


Figure 4.3: Architecture of a WFTT network

use in the WFTT protocol.

As introduced, the WFTT was inspired by the FTT paradigm [159]. In this sense, it employs a FTT-like master-multislave communication model in which a master station synchronizes the transmissions from slave stations using a trigger packet. The trigger packet defines which stations (and when) are allowed to transmit in a time interval named *Elementary Cycle* (EC). The critical station (CST1) assumes the role of the WFTT master, since it is the only station with potential to perform data transmissions with bounded communication delay, a key operational flexibility requirement of the FTT paradigm. Provided that the *bandjacking* MAC technique was demonstrated to support dependable communications, it is foreseeable that its transmission scheme will result in critical transmissions characterized by a highly deterministic delay. Hence, the trigger packet was mapped to the *bandjacking*'s critical packet. In this sense, it is preceded by a *black-burst* sequence that clears the medium from "alien" station transmissions. The standard stations SST2, SST3 and SST4 play the role of the WFTT slaves, i.e., they only act upon receiving a trigger packet, decoding it and checking if they are producers of some message in the corresponding elementary cycle.

The preference for the FTT paradigm in the development of a real-time wireless communication protocol was mainly motivated by its architecture and temporal organization. This option allowed inheriting the paradigm's properties and their validation, besides benefiting from the existing operation and implementation know-how. Provided the centralized nature of the FTT paradigm, the use of a single critical station as the master seems the natural option considering the overall cost and functionality. The design of a commercially viable network hardware architecture must endeavor for making the most pervasive devices cost less so that the network cost is mostly dependent on these devices. On account

that the critical station incorporates the PNS, which makes it much more expensive than standard stations, its use for the role of the central station (master) seems reasonable. In consideration that the trigger packet corresponds to a critical communication, only the critical station has the functionality required to support the role of the master. Therefore, in an architectural perspective, the use of the critical station as the WFTT master directly emerges from the requirements of the FTT paradigm.

Regarding the temporal organization, the FTT protocol is particularly suited for supporting different types of traffic, namely, event and time triggered. Since the WFTT protocol aims at supporting a wide variety of applications, possibly encompassing different communication requirements, the flexibility and timing guarantees conveyed by the FTT protocol seem a good base for building an efficient wireless communication protocol. In this sense, the protocol can be designed to inherit the ability of adjusting to different operational conditions, maximizing the use of the available resources in any instant (to grant the required levels of quality of service) and allowing reconfiguring the communication parameters on-line.

The described WFTT protocol deals only with the low level communications' implementation that guarantees the first operational flexibility requirement defined in [164]: bounded communication delays. In other words, WFTT protocol herein presented exclusively addresses the requirement of providing low-latency deterministic wireless communications. The support for the remaining requirements of the FTT paradigm is outside the scope of this work. Nevertheless, provided the similar characteristics, the mechanisms developed for the FTT paradigm should be easily extended to the WFTT protocol.

Elementary Cycle

The WFTT elementary cycle (EC) is a period of time characterized by a predefined duration that encompasses several communication phases (windows), as depicted in Figure 4.4. Provided that the TP is considered a critical transmission in the WFTT protocol implementation, it is preceded by a *black-burst* interference sequence that guarantees the medium availability at its end, as demonstrated in the last chapter. Afterwards, the TP is transmitted immediately, synchronizing the communication stations participating in the network. Besides providing the means to synchronize the WFTT network, the TP also conveys detailed information about the synchronous messages that must be transmitted by the slave stations in the current ECs and, at the same time, provides information about the window timing bounds of the EC. This master-multislave scheme, employing

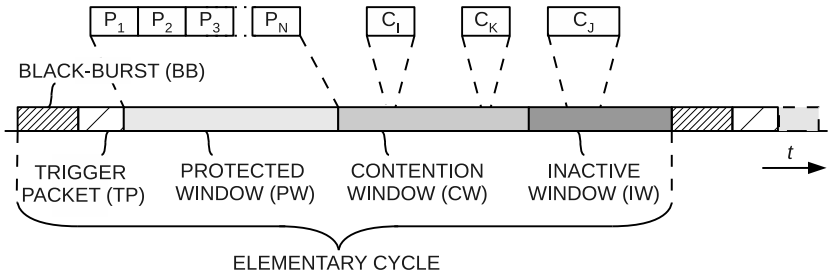


Figure 4.4: Elementary Cycle diagram

a single message to control the behavior of multiple slave stations, presents a significant overhead improvement when compared to common master-slave protocols, leading to an overall higher network throughput.

A temporal sequence of communication phases (windows) follows the TP transmission: *protected* window (PW), *contention* window (CW) and *inactive* window (IW). The WFTT protocol defines specific access rules for each timing window. The PW envisages the transmission of the real-time packets explicitly scheduled in the preceding TP, in this case, $P_1, P_2, P_3 \ldots P_N$. These transmissions are made almost "back together" to maintain the medium occupied and to prevent neighbor "alien" contention-based technologies from having the chance of finding the medium idle and initiating a transmission. This continuous occupation of the medium is achieved both by injecting *black-burst* interference between slave transmissions and protective interference simultaneously to the slave transmissions in the low and high "narrowbands" (represented in Figure 3.2).

The protected window is the EC interval where collisions must not occur, neither between WFTT transmissions nor with contention-based "alien" technologies. This contributes to a higher network throughput, energy preservation and quality of service. Since the transmission medium is maximally used, no transmissions are wasted due to collisions and the medium access latency is minimum because of the enforced TDMA scheme. The WFTT protocol energy preservation focus is on the slave stations, which can be mobile and operate on batteries, thus having an autonomy highly coupled with their power consumption. Although the master station broadcasts interference patterns that require a significant amount of energy, because it is designed to be mains powered, it has loose operational requirements regarding energy preservation.

The CW establishes a period of time for asynchronous traffic where WFTT standard stations may compete for the medium using native contention-based mechanisms, for example, the IEEE 802.15.4 CSMA/CA scheme. Besides the WFTT standard stations, "alien"

stations using the same communication technology can dispute the usage of the medium during this period, as long as they operate on the same channel. However, since protective interference is issued by the master's PNS during this window, "alien" stations using a different (contention-based) technology will be hindered from disputing the medium. Figure 4.4 documents an example of two contention-based transmissions: C_I and C_K. During the CW all stations of WFTT network can freely request the transmission of packets. However, only those which can be concluded within the window are allowed to proceed. Otherwise, the requests that could not be served within the window are queued and submitted again in the CW of the subsequent EC. The contention window provides a moderate network throughput and a best-effort quality of service as result of the periods of inactivity introduced by the contention scheme and the uncertainty of the request load offered during this interval. The energy preservation during this interval is also high, as the contention mechanism avoids wasting energy on collisions. This further enforced by the use of protective interference, which prevents "alien" stations from initiating transmissions as a result of perceiving the medium idle when WFTT low-power transmissions are ongoing.

The IW completes the EC, establishing an interval of time where standard stations with demanding autonomy requirements can switch to a mode of reduced power consumption by turning off their radio-frequency transceivers until the next TP, for example. During this period, "alien" stations may access the medium and perform transmissions. The transmission of an "alien" packet in the inactive window is represented as C_J in Figure 4.4. In addition to establishing a mechanism to enhance the autonomy of standard stations with stringent low-power requirements, this timing window contributes to enforcing a fair use of the medium by allowing transmissions from other technologies without interference from the WFTT network.

Packet Structure

The structure of a WFTT packet is presented in Figure 4.5. The *trigger*, *real-time* and *contention* packets are built using standard communication MAC packets, encapsulating the specific information in the frame's payload. The fields *type of packet* and *flags* are common to all packet types. The *flags* field includes the sequence number. Each packet has a distinct *payload* according to its type. The *trigger packet* specifies the *number of real-time packets* that will be conveyed during the PW, the specific information of each one (*identification* and *offset*), the *contention window duration* and the *elementary cycle duration*.

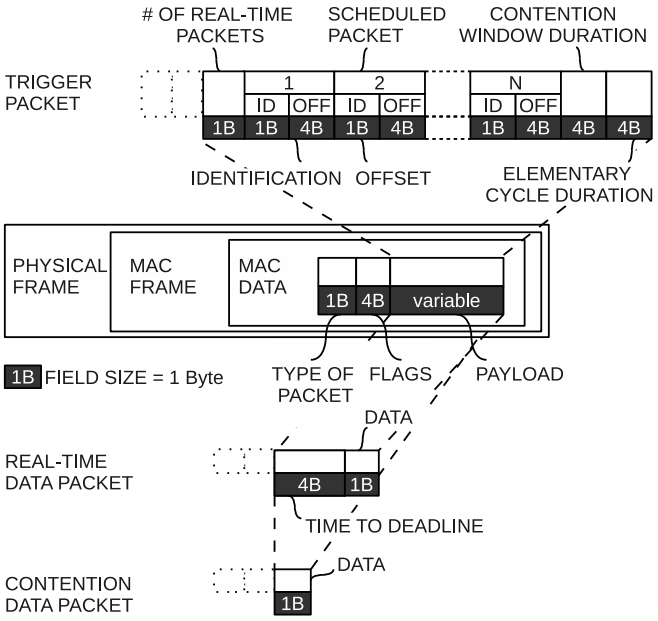


Figure 4.5: Structure of WFTT packets

In the following analysis, for simplification/compactness purposes, it is assumed that the PW real-time transmissions have a constant length. However, variable sized real-time transmissions can be supported by including in the TP an additional field defining each transmission's duration. Since the trigger packet specifies the offsets of all the transmissions in this window, the contention window duration and the elementary cycle duration, slave stations can easily determine the time bounds where they must be active (protected and contention windows) or switch off their communication transceivers (inactive window). Although real-time and contention data packets have a similar structure, the former has an additional field specifying the *time to deadline*, which will be used to determine the temporal validity of the data. Besides documenting the field organization of valid WFTT packets, Figure 4.5 also presents the size of each field, which is important to determine the duration of each packet type.

Temporal isolation

The temporal isolation of the time windows within each WFTT EC is ensured by different mechanisms, depending on the interval being considered. The temporal isolation of the PW with regards to the CW is secured both by the intrinsic characteristics of the WFTT protocol and the discipline enforced on the slave stations. As to the first, the medium will never be perceived as idle during the PW due to the scheduled real-

time transmissions and to the interference sequences that protect them. Hence, even if contention-based WFTT slaves would attempt to perform transmissions during this period, they would be forced to postpone them to a later time due to finding the medium busy. With reference to the discipline enforced on the slave stations, the traffic transmitted during the PW must be explicitly authorized by the master station in the preceding TP and cannot be carried out beyond the specified temporal limits. Therefore, only the slave stations that were authorized to produce information in the EC's protected window can perform transmissions on the designated slots. In the future, fault-tolerance mechanisms similar to the ones developed for FTT-CAN and denoted as bus guardians [165] (in this case media guardians) can possibly be introduced in the slaves to guarantee that they will respect the time bounds imposed by the master.

The isolation between the CW and the IW is established by ensuring that the slave stations respect the temporal bounds of the CW, as defined by the preceding TP. In this sense, WFTT slaves are only enabled to dispute the medium when the contention window begins and can only request new transmissions if they are guaranteed to be concluded within the bounds of the CW. Otherwise, if a transmission is submitted but it extends beyond the limits of the CW, it is queued and resubmitted in the CW of the subsequent EC, where the process repeats. The instant in which the transmission is allowed can be used to establish priorities among contention based transmissions. For instance, transmission requests originating in resubmission of packets are performed at the beginning of the CW, thus increasing their chances of success.

The separation between the IW and the following PW is ensured by the WFTT protocol operation itself. First, no WFTT transmissions are allowed to be performed during this period, as WFTT slave stations with stringent energy consumption requirements will be held in a low-power state, which hinders them from receiving any packets during this interval. Second, slave stations that are being kept on a low power state are triggered to wake up just before the end of the *black-burst* noise sequence, on time to receive the TP that schedules the following EC. Third, in order to comply with the WFTT protocol timings, all slave stations must actively listen to the trigger packet transmission performed by the master. Holistically, these requirements guarantee that slave stations are not able to perform any transmissions from the end o the CW to the beginning of the PW.

4.2.2 Operation

Figure 4.6 presents an example of a WFTT EC that will be used to describe the protocol's operation and derive the expressions ruling its timing behavior. As documented, the EC encompasses a sequence of protected, contention and inactive windows, which are preceded by a critical data transmission composed by a *black-burst* sequence and a trigger packet. As shown, different visual patterns are used to differentiate the intervals of time pertaining to the packet and interference transmissions. The identified parameters are detailed in Tables 4.1, 4.2 and 4.3 according to its fixed or variable nature[1]. Five parallel individual timelines are used to better represent the transmissions from both the PNS (PNS1) and the slave stations (SST1, SST2, SST3 and SST4) over time. In the supplied example, the TP schedules three data transmissions that are carried on during the PW. In addition, the represented timelines do not show the transmission's propagation delays, as these can be considered negligible for personal area network communication ranges.

Table 4.1: Fixed parameters

Name	Description
D_{IS}	Delay of starting an Interference Sequence (*IS*), measured from the instant of the request (by SST1) to the instant of its effectiveness (at PNS1)
D_{PD}	Delay of starting a Protected Data (*PD*) transmission
D_{TP}	Delay of initiating a Trigger Packet (*TP*) transmission, measured from the instant of its request to the instant of its effectiveness
L_{BIS}	Length of a *Bandjacking* Interference Sequence (*BIS*)
L_{IFS}	Length of the Inter Frame Space (*IFS*) between two consecutive data transmissions
L_{ISI}	Length of the Inter Frame Space Interference (*ISI*) sequence
L_{OI}	Length of the Overhead Interference (*OI*) sequence

In the following, a description of the annotated elements of Figure 4.6 is presented. Explicatively, at instant t_1, the master reconfigures the interference profiles that will be used in the following EC(s). This reconfiguration is conducted during the inactive window and allows the master to change the parameters of the PNS's interference profiles according to different requests (e.g., change channel or power), thus supporting the on-line adjustment of the WFTT network to different working conditions. The reconfiguration is only accepted if it can be concluded within the timespan of the IW. Otherwise, it is queued and postponed to the IW of the following EC.

At instant t_2 the CTS1 initiates a critical transmission by triggering the propagation of

[1]In this dissertation the jitter is defined as the variable component of a delay, independently of the event nature (periodic or aperiodic) being measured.

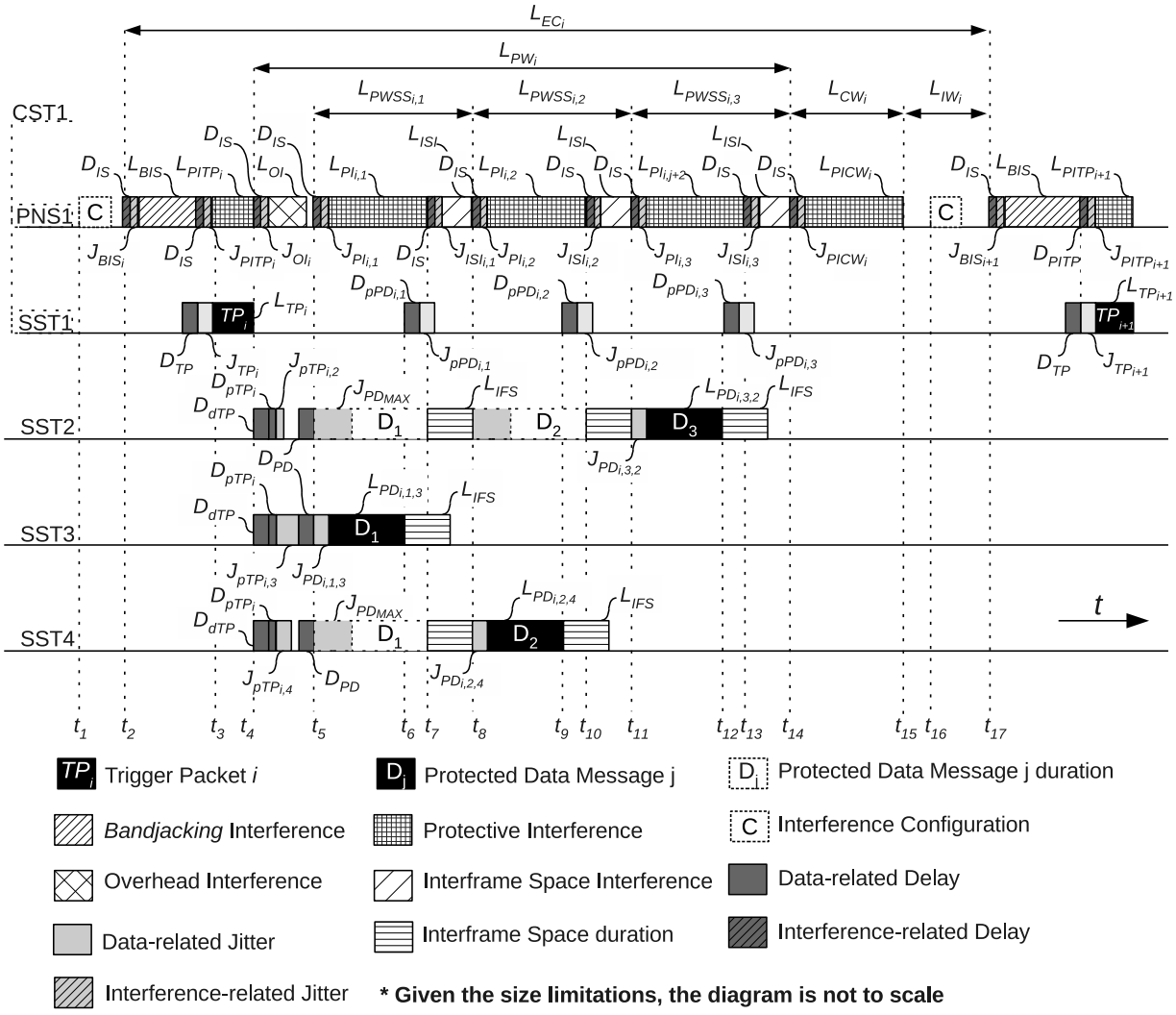


Figure 4.6: WFTT EC timeline

a *black-burst* (L_{BIS}) by the PNS1. This transmission is affected by a D_{IS} delay and a J_{BIS_i} jitter. The first parameter represents the constant part of the activation delay (minimum). The second represents its variable range. The approach of segmenting the delay parameters in two separate contributions (constant and variable) for the events depicted in Figure 4.6 is adopted in the remaining of this dissertation.

In order to guarantee that all alien stations are in a backoff state immediately after the end of the *black-burst* sequence, its duration is selected according to the maximum transmission duration foreseeable for any "alien" packet m (see Theorem 3.2.5). In this sense, Equation 4.1 defines the required *black-burst* length as a function of the "alien" data packet duration (L_{AD_m}).

Table 4.2: Variable parameters: delays, lengths and quantities

Name	Description
D_{dTP}	Delay of the transceiver to signalize the reception of a packet with an interrupt after the energy broadcast in the medium has ended
$D_{pPD_{i,j}}$	Delay of processing the j^{th} Protected Data transmission in the i^{th} EC and request an interference commutation
D_{pTP_i}	Delay of processing the TP associated to the i^{th} EC
$D_{rTPi,k}$	Delay of the k slave station to respond to the TP in the i^{th} EC. Includes the worst case latency that any slave station takes to process the TP, load a real-time data packet and switch the transceiver to the transmit state
L_{AD_m}	Length of a given "alien" packet m
L_{CW_i}	Length of the Contention Window (CW) in the i^{th} EC
L_{EC_i}	Length of the i^{th} elementary cycle (EC)
L_{IW_i}	Length of the Inactive Window (IW) in the i^{th} EC
$L_{PD_{i,j}}$	Length of the j^{th} Protected Data (PD) transmission (in the protected window) of the i^{th} EC
$L_{PI_{i,j}}$	Length of the Protective Interference (PI) sequence for the j^{th} data transmission (in the protected window) of the i^{th} EC
L_{PICW_i}	Length of the Protective Interference sequence in the Contention Window ($PICW$) of the i^{th} EC
L_{PITP_i}	Length of the Protective Interference sequence for the TP ($PITP$) transmission of the i^{th} EC
L_{PW_i}	Length of the Protected Window (PW) in the i^{th} EC
$L_{PWSS_{i,j}}$	Length of the j^{th} Slave Slot in the Protective Window ($PWSS$) of the i^{th} EC
L_{TP_i}	Length of the TP transmission in the i^{th} EC
N_{PPW_i}	Number of Packets in the Protected Window (PPW) of the the i^{th} EC

$$L_{BIS} = max\left(L_{AD_m}\right), \forall m \in \mathbb{N}_1 \tag{4.1}$$

Provided that the TP transmission has a duration of L_{TP_i} and experiences a D_{TP} constant delay, in order to guarantee that it begins with the shortest possible delay after the end of the *black-burst*, its transmission is initiated at instant $t_3 - D_{TP}$, becoming effective within the J_{TP_i} timing window (TP jitter). The duration of the TP transmission can be modified between consecutive ECs, since the number of scheduled packets in the PW can be updated on-line. However, if the EC period is to be kept constant, the duration of the CW or IW has to be updated to compensate for both the TP and PW modifications.

As shown, the TP is protected by an interference sequence in the low and high "narrow-bands" (L_{PITP_i}), herein named *Protective Interference* (PI), which is synthesized simultaneously to the TP transmission just after the *black-burst* interference sequence. When the TP transmission ends (at instant t_4), the SST1 signals the PNS1 to stop the protective in-

Table 4.3: Variable parameters: jitter (delay variation)

Name	Description
J_{BIS_i}	Jitter of initiating *Bandjacking* Interference Sequence (*BIS*) in the i^{th} EC
$J_{ISI_{i,j}}$	Jitter of initiating an Inter Frame Space Interference (*ISI*) sequence after the the j^{th} data transmission in the protected window of the i^{th} EC
$J_{ISI_{MAX}}$	Maximum jitter of initiating an Inter Frame Space Interference
J_{OI_i}	Jitter of initiating an Overhead Interference (*OI*) sequence in the i^{th} EC
$J_{PD_{i,j,k}}$	Jitter of the k slave station to start the j^{th} Protected Data (*PD*) transmission (in the protected window) of the i^{th} EC
$J_{PD_{MAX}}$	Maximum jitter (considering all slaves) of initiating a Protected Data (*PD*) transmission
$J_{PI_{i,j}}$	Jitter of initiating the Protective Interference (*PI*) sequence associated to the j^{th} data transmission in the protected window of the i^{th} EC
J_{PICW_i}	Jitter of initiating a Protective Interference sequence in the Contention Window (*PICW*) of the i^{th} EC
J_{PITP_i}	Jitter of initiating a Protective Interference sequence for the TP (*PITP*) of the i^{th} EC
$J_{pPD_{i,j}}$	Jitter of processing the j^{th} Protected Data transmission in the i^{th} EC and request an interference commutation
$J_{pPD_{MAX}}$	Maximum jitter of processing a Protected Data transmission by the master station
$J_{pTP_{i,k}}$	Jitter of the k slave station to process the Trigger Packet of the i^{th} EC
J_{TP_i}	Jitter of initiating a TP transmission in the i^{th} EC

terference associated to the TP transmission. As any other transmissions, it encompasses a given activation delay (D_{IS}) and an associated jitter (J_{PITP_i}). The WFTT protocol design supporting Figure 4.6 assumes that all types of interference produced by the PNS are characterized by a constant activation delay expressed by D_{IS} and are switched off instantly, i.e., can be disabled with a negligible latency.

In the following subsections, a detailed description of the operation of the WFTT protocol in the *protected*, *contention* and *inactive* windows is provided.

Protected Window (PW)

The i^{th} protected window is characterized by a L_{PW_i} length. This window begins at instant t_4 with an overhead delay that aims at providing enough room for the slaves to receive the TP, process it and become ready to respond with a real-time transmission, when required. The overhead delay, thus, has three contributions: TP detection delay, the TP response delay and the real-time packet trigger delay. The first corresponds to the latency of the transceiver to signalize the reception of a packet with an interrupt after the energy broadcast ended in the medium. This latency (D_{dTP}) is assumed to be constant and, thus, not coupled with the TP length. The TP response delay includes the worst case

latency that any slave station takes to process the TP, load a real-time data packet in the transmission buffer and switch the transceiver from the receive to the transmit state. The overall TP response delay required by a generic slave station k to conduct these tasks in the elementary cycle i is given by Equation 4.2.

$$D_{rTPi,k} = D_{pTP_i} + J_{pTP_{i,k}}, \forall i,k \in \mathbb{N}_1 \tag{4.2}$$

The TP may have different lengths on different elementary cycles. Therefore, the delay experienced by slave stations to process the TP can also change between ECs. Moreover, the delay of loading the real-time packet to the transceiver and switching it to the transmit state is assumed to be constant, because the packet length was also assumed to be constant regardless of the PW slot where it is transmitted. The jitter associated to the aforementioned tasks is variable, i.e., it will change depending on the EC and slave station performing them. Henceforth, the constant part of the TP response delay is modeled by the D_{pTP_i} parameter while its variation is expressed by $J_{pTP_{i,k}}$, where k represents a specific slave station receiving the TP.

The real-time packet trigger delay also comprises two components: one fixed and one variable. The first models the constant latency that any real-time packet transmission encompasses and it is denoted by D_{PD}. The second, is the associated transmission jitter expressed by $J_{PD_{i,j,k}}$, where i refers to the i^{th} EC, j to the j^{th} PW time interval and k to the station k.

The overhead delay length (L_{OH}) enforced after the TP transmission is described by Equation 4.3 and comprises the TP detection delay, the worst case TP response delay and the constant contribution to the real-time packet trigger delay. In order to keep the medium occupied during this period and hinder "alien" contention-based stations from initiating transmissions, the master (CST1) uses the PNS1 to synthesize "wideband" interference during a L_{OI} interval, as defined by Equation 4.4. This interference sequence is named *Overhead Interference* (OI) and experiences an activation delay and jitter given by D_{IS} and J_{OI_i}, respectively.

$$L_{OH} = D_{dTP} + max(D_{rTPi,k}) + D_{PD}, \forall i,k \in \mathbb{N}_1 \tag{4.3}$$

$$L_{OI} = L_{OH} - (D_{IS} + max(J_{OI_i})) - max(J_{pTP_{i,k}}), \forall i,k \in \mathbb{N}_1 \tag{4.4}$$

As shown, the L_{OI} interval corresponds to the overhead delay L_{OH} reduced by two safety margins: one of $(D_{IS}+max(J_{OI_i}))$ and the other of $max(J_{pTP_{i,k}})$. The first compensates the overhead interference activation latency and jitter. The second counteracts the maximum trigger packet processing jitter. By enforcing both time margins, it is possible to avoid the overhead interference from extending beyond its designated bounds and compromise the real-time transmission occurring in the subsequent PW slot.

Provided that the transceiver takes a given amount of time between the protected packet trigger event and its effective transmission on the medium, the constant (predictable) portion of this delay can be anticipated. Hence, when the master station computes the offsets that are used by the slave stations to trigger the scheduled transmissions, it subtracts a D_{PD} amount of time in order anticipate the trigger event so that the resulting gap between the end of the overhead interval and the beginning of the first transmission is minimized.

As scheduled by the TP, slave stations initiate transmissions in specific instants and over disjoint periods of time, called slots (e.g., $L_{PWSS_{i,j}}$). The master employs Equation 4.5 to compute the activation offsets of the slave stations ($ao_{PD_{i,j,k}}$), which prevent collisions among protected transmissions. Because the slave stations only detect the TP at instant $t_{4'} = t_4 + D_{dTP}$, the first transmission must be scheduled to be triggered at the relative instant $max(D_{rTPi,k})$, in order to guarantee that the effective transmission suffers the least possible delay to become effective in the beginning of the first slot.

$$ao_{PD_{i,j,k}} = \begin{cases} max(D_{rTPi,k}) & j = 1 \\ max(D_{rTPi,k}) + \sum_{m=1}^{j-1} \left(J_{PD_{MAX}} + L_{PD_{i,m}} + L_{IFS} \right) & j \geq 2 \\ \forall i, k \in \mathbb{N}_1 \wedge j \in [1...N_{PPW_i}] \end{cases} \quad (4.5)$$

where:

$$j_{PD_{MAX}} = max\left(J_{PD_{i,j,k}}\right), \forall i, k \in \mathbb{N}_1 \wedge j \in [1...N_{PPW_i}] \quad (4.6)$$

This implementation approach is different from the one described in [158] due to the fact that, instead of locally computing the offset of their transmissions on the PW, slaves simply obtain that information directly from the TP, in order to shorten their response time. Hence, upon receiving the TP and if they are producers of a packet in the PW, slave stations immediately load a timer with the offset value read from the TP. This timer will expire at the instant where the packet transmission will be triggered, as calculated by the master station.

At instant $t_{4''} = t_4^{'} + max(D_{rTPi,k})$, SST2 begins the process of transmitting a real-time data packet whose duration is represented by $L_{PD_{i,j}}$. When this packet transmission ends (at instant t_6), the master station (CST1) processes it with a $D_{pPD_{i,j}}$ delay and $J_{pPD_{i,j}}$ jitter. At instant t_5 the master begins the PI transmission with a D_{IS} delay and a $J_{PI_{i,j}}$ jitter. This transmission extends up until the t_7 instant, corresponding to the worst case delay for the real-time packet transmission completion ($J_{pPD_{MAX}} + L_{PD_{i,j}}$).

In order to ensure that all slave stations are able to successfully receive the real-time transmissions, it is necessary to establish an interval between real-time packet transmissions equal to the Inter Frame Space (IFS) of the data technology being employed. During this period, the master synthesizes "wideband" IFS Interference (ISI) to maintain the medium occupied. This interference sequence is influenced by a given delay and jitter denoted by D_{IS} and $J_{ISI_{i,j}}$, respectively. Hence, at instant t_7, the master requests a ISI interference sequence with a length is determined according to Equation 4.7.

$$L_{ISI} = L_{IFS} - (D_{IS} + J_{ISI_{MAX}}) \tag{4.7}$$

where

$$J_{ISI_{MAX}} = max\left(j_{ISI_{i,j}}\right), \forall i \in \mathbb{N}_1 \wedge j \in [1...N_{PPW_i}]$$

At instant t_8, immediately after the end of the ISI, the CST1 automatically switches the interference profile back to the PI again, in order to protect the incoming real-time data transmission from SST4, which is triggered at instant $t_{8'} = t_8 - D_{PD}$. As documented, the real-time data transmission jitter ($J_{PD_{i,j,k}}$) affects its activation offset within the PW scheduled slot. Two scenarios arise when considering the delay (D_{IS}) and jitter ($J_{PI_{i,j}}$) of initiating a PI transmission:

1. The overall protection interference activation delay is smaller than the jitter of initiating the protected data transmission (D_{IS}+$J_{PI_{i,j}}$) < $J_{PD_{i,j,k}}$. This seems to be the most common scenario since the PNS can exhibit a time response that only depends of the activation delay of a power amplifier. Hence, it is assumed that the sum of D_{IS} and $J_{PI_{i,j}}$ are negligible. In this case, the PI begins before the real-time transmission and the channel used by the real-time transmissions will be idle during the $J_{PD_{i,j,k}}$ jitter. This scenario may be further analyzed and segmented according to the type of "alien" stations that contend for the medium.

(a) Stations employing the same real-time data technology on the same channel. In this case, if the $J_{PD_{i,j,k}}$ jitter is larger than the minimum IFS or CCA intervals employed by "alien" stations, they can initiate a transmission during this period and, consequently, compromise the real-time transmissions of the WFTT's PW.

(b) Stations employing a different data technology with a higher bandwidth overlapping the channel used for real-time transmissions. The PI keeps the data's lateral channels occupied, generating enough energy to prevent these stations from performing transmissions during the PI sequence ($L_{PI_{i,j}}$).

(c) Stations employing a different data technology with a smaller bandwidth overlapping the channel used for real-time transmissions. This case is similar to the first, where the PI is not capable of making the "alien" stations perceive the medium as busy. Hence, "alien" stations characterized by IFS or CCA intervals shorter than the $J_{PD_{i,j,k}}$ jitter may compromise the real-time transmissions of the PW.

2. The overall protection interference activation delay is greater or equal to the jitter of initiating the protected data transmission (D_{IS}+$J_{PI_{i,j}}$) $\geq J_{PD_{i,j,k}}$. This case will be rare due to the responsiveness provided by the PNS hardware architecture. However, because the real-time transmission jitter is variable, it can (sporadically) assume small values and become shorter or equal to the overall PI activation delay. In this case, assuming that the protective interference activation delay is negligible (as in point 1), the period of time in which the medium is idle will be smaller than the IFS or CCA of any "alien" station contending for the medium. Therefore, the WFTT schedule in the PW will not be compromised by unauthorized transmissions.

In both scenarios, if the real-time packet jitter ($J_{PD_{i,j,k}}$) is small, the transmission will occur at the beginning of the PW slot, leaving a time interval near the end of the slot that will be secured by protective interference. The remaining events in the PW (concluded at instant t_{14}) follow the approach described for the first protected data transmission.

Contention Window (CW)

As introduced, the contention window aims at providing an interval of time for asynchronous traffic. As discussed in Section 2.3, technologies using higher bandwidths and higher power levels have a significant impact on the timeliness and reliability of co-located

low-power communications operating on the same region of the spectrum. In this sense, the WFTT CW was designed to enforce an interval of time where such "alien" technologies are hindered from compromising asynchronous traffic performed by low-power WFTT slave stations.

Figure 4.6 depicts a CW that is initiated at instant t_{14} and has a length of L_{CW_i}. In this window, in order to protect the WFTT contention-based transmissions against interference from "alien" stations employing technologies with larger bandwidths, the master station occupies the medium in the low and high interference "narrowbands", as documented in Figure 3.2. This CW protective interference (PICW) is affected by a given activation delay (D_{IS}) and jitter (J_{PICW_i}). During the CW, slave stations may perform transmissions using standard contention-based methods as long as their duration does not extend beyond the CW bounds defined by the master station. If this occurs, the transmission is queued and a retransmission attempt is performed at the beginning of the CW of the subsequent EC.

Inactive Window (IW)

The inactive window follows the CW. It begins at instant t_{15} and has a duration of L_{IW_i}, which extends until the next *bandjacking* access. This interval was included in the WFTT EC to establish a period of time reserved for low-power slaves to sleep and for "alien" stations to conduct transmissions and avoid starvation, while operating over the same region of the spectrum. The support for a period of inactivity in the WFTT protocol specifically addresses applications with stringent low-power requirements, i.e., applications with low duty cycle requirements, but demanding a high level of dependability and timeliness. Also, the existence of a "silent" WFTT interval promotes a fairer sharing of the communication channel with "alien" contending stations.

In a basic WFTT star network such as discussed so far, no slave WFTT transmissions are allowed to take place during the IW. This occurs because slave stations with strict autonomy requirements switch to a low-power operation mode during this period to extend their autonomy. Among other actions, these stations can slow down the frequency of the MCU, turn off unnecessary MCU peripherals and shut down the communications transceiver, thus hindering them from receiving packets. In the course of this period, the master station may reconfigure its interference patterns and perform calculations (e.g., packet scheduling) regarding the following EC.

Although beyond the scope of this dissertation, it is possible to use the inactive period to enable communications among WFTT star networks such as, for example, in the WIA-

PA protocol addressed in Section 2.2.9. In this case, the inactive window would encompass protected transmissions among WFTT star networks. Since the master is permanently active, it can participate in data exchanges during the inactive window.

4.2.3 The Hidden Node Problem

The hidden node problem is a well-known phenomenon affecting wireless networks. This problem occurs when a sink station is within the range of a set of source stations, which are not fully "visible" to each other. Therefore, two source stations hidden from each other may attempt to transmit simultaneously to the sink, thus causing a packet jamming at the sink and the loss of both transmitted packets.

As presented, the WFTT protocol was based on the *bandjacking* technique for guaranteeing the deterministic access of the master's trigger packet to the medium. Furthermore, the protocol also relies on the use of *black-burst* and protective interference to prevent "alien" stations from accessing the medium and impairing real-time transmissions performed by slave stations. The existence of hidden master, slave or "alien" stations represents a challenge to the determinism conveyed by the WFTT protocol. In this sense, several possible scenarios of interaction between wireless stations were identified and characterized.

1. The "alien" station is out of range of the interference synthesized by the master station as well as of the real-time transmissions performed by the slave stations. The "alien" station transmissions are out of range of all the WFTT stations.

2. The "alien" station is out of range of the interference synthesized by the master station as well as of the real-time transmissions performed by the slave stations. The "alien" station transmissions are in range of some/all slave stations.

3. The "alien" station is in range of the interference synthesized by the master station but not of all real-time transmissions performed by the slave stations. The "alien" station transmissions are in range of some/all WFTT stations. In this scenario, two occurrences are possible:

   (a) The bandwidth employed in real-time transmissions is similar to the bandwidth used by the "alien" technology.

   (b) The bandwidth employed in real-time transmissions is much smaller than the bandwidth used by the "alien" technology.

4. The "alien" station is in range of the interference synthesized by the master station as well as of all real-time transmissions performed by the slave stations. The "alien" station transmissions are in range of all WFTT stations.

Scenarios 1 and 4 are not likely to pose any issues regarding the hidden node problem, provided that "alien" stations are distant enough from the WFTT network (master and all slave stations) to cause any interference or are able to both sense the real-time communications and the synthesized *black-burst* and protective interference, thus limiting their access to the medium. Following the same trend, scenario 3(b) should also be free from hidden node impairments since the protective interference will prevent higher bandwidth "alien" stations from initiating transmissions. Conversely, the lack of impact of the protective interference in avoiding "alien" transmissions in scenario 3(a) can result in the corruption of real-time packets sent by hidden slave stations. This scenario should not occur frequently since it is typically characterized by the use of the same technology by the WFTT network and by the "alien" stations. Hence, the communication range should be similar in both directions (slave $\rightarrow$ "alien" station and vice versa). Nevertheless, one possible solution for this problem combines the configuration of the WFTT slave stations to employ a higher transmission power and the adoption of a physical region surrounding the WFTT network where no "alien" stations can be deployed. The first proposal extends the slave's physical reach, thus contributing to their sensing by the "alien" stations. The second reduces the impact of the "alien" transmissions on the WFTT network, i.e., it increases the Signal-to-Interference Ratio (SIR) of the WFTT transmissions.

Scenario 2 is also prone to the hidden node problem, as an "alien" station may interfere with a real-time transmission from a given slave station. Although there are many solutions to mitigate the hidden node problem in wireless networks [166], none of them seems directly applicable to the specific operation of the WFTT protocol, since the main vulnerable scenario occurs when the master station is hidden from the "alien" stations. Provided that WFTT and "alien" stations do not communicate in the same network, control packets cannot be exchanged to setup a collision free environment. A possible way of avoiding interference from "alien" stations is the adoption of a special WFTT station that simply produces *black-burst* and protective interference synchronized with the master station. This special station would be placed in the physical boundaries of the WFTT network, where the master's interference transmissions are hidden from local "alien" stations. The hidden node problem is still an open issue in wireless networks and a complete solution is beyond the scope of this work.

4.3 Analytical Study

The WFTT protocol is studied under two perspectives: implementation feasibility and timeliness. The objective of the former is to identify the key intervals of time in which the protocol operation can be compromised. Furthermore, this section also establishes the allowed operation bounds for the parameters that represent such intervals, allowing to determine the feasibility of implementing and deploying the WFTT protocol in real nodes. The timeliness study of the WFTT protocol is focused on analyzing two types of traffic: synchronous and asynchronous. This timeliness characterization provides the necessary background information to allow estimating the delay and jitter experienced by WFTT packets in both traffic scenarios.

4.3.1 Feasibility

This section presents a WFTT timing analysis focused on the challenges of implementing the protocol with low resource slave stations. This analysis is derived from the WFTT EC represented in Figure 4.6 and encompasses a coexistence study between the WFTT protocol and "alien" stations employing different communication technologies. Within this scope, two complementary interference scenarios are addressed:

1. "Alien" technologies interfering over a bandwidth much wider than the data band (i.e., $BW_{data} \ll BW_{alien}$). In this case, without loss of generality, it is assumed that WFTT stations employ the IEEE 802.15.4 technology to perform data transmissions and that "alien" stations are compliant with the IEEE 802.11 standard technology. The study of this scenario is motivated by the fact that the IEEE 802.11 protocol is, currently, one of the most widely deployed (and potentially harmful) wireless local area network technology, as described in Section 2.3.3;

2. The data technology has a bandwidth similar to ($BW_{data} \approx BW_{alien}$) or higher than ($BW_{data} > BW_{alien}$) the "alien" technology. As before, the WFTT was assumed to employ the IEEE 802.15.4 technology for its data transmissions. Regarding "alien" transmissions, due to its adoption for supporting wireless personal area networks (e.g. ZigBee), the IEEE 802.15.4 was selected as the contender technology. This option was motivated by the coexistence conclusions presented in Sections 2.3.1 and 2.3.2.

The timing analysis in both scenarios is supported in several assumptions regarding the operation of both the WFTT and "alien" stations. As described, besides using the

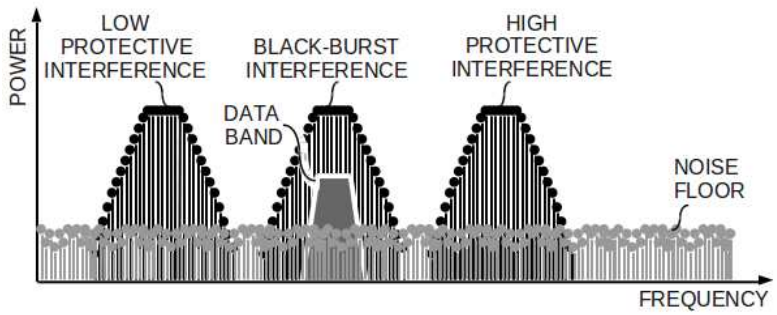


Figure 4.7: WFTT spectrum occupation illustration

IEEE 802.15.4 protocol for the WFTT data transmissions, the existence of two different contending technologies (IEEE 802.11 and IEEE 802.15.4) is considered. In both cases, the medium access is conducted according to the corresponding specification. Hence, transmissions can only begin when the medium is perceived idle for a period of time longer than the CCA or IFS defined by the contending technology. Furthermore, it is assumed that the interfering technologies employ their nominal power levels and perform transmissions on a channel overlapping the WFTT data transmission channel.

Regarding the hardware of the WFTT stations, the master's communication controller (CC1) is assumed to have enough processing power to run the WFTT services in a timely fashion. Hence, it is accepted that the PNS complies with the following assumptions:

i) *The interference signals synthesized by the master are compliant with the bandwidth profiles represented in Figure 4.7*;

   This is a reasonable assumption since the PNS presented in Section 3.4 can be configured to synthesize interference in different channels with a bandwidth similar to the one employed by standard IEEE 802.15.4 stations ($BW_{data} \approx BW_{interference}$). Despite the fact that the presented COTS-based implementation is not able to conduct simultaneous transmissions in two different channels, the hardware architecture can be modified to include more transceivers and, therefore, to enable the transmission of interference in two simultaneous channels. The protective interference is transmitted simultaneously in both the low and high channels, which are two channels apart (IEEE 802.15.4) from the WFTT data channel. The *black-burst* interference occupies an overlapping spectrum region, i.e., it is centered on the channel used to perform the WFTT data transmissions.

ii) *Interference signals are compliant with the EU and USA regulations for transmitting*

in the 2.4 GHz ISM band;

The legal limit to perform transmissions employing Direct Sequence Spread Spectrum (DSSS) signals in the 2.4 GHz ISM band is 100 mW (20 dBm) of Equivalent Isotropically Radiated Power (EIRP) in European countries (standard ETSI EN 300 328) and 1W (30 dBm) of Peak Conducted Output Power in the United States of America (regulation FCC part 15.247 and 15.249). Therefore, it is fair to assume that because the hardware architecture of the COTS-based PNS employs certified modules, it can be used to generate DSSS signals complying to these limits.

iii) *The synthesized interference signals are effective in hindering "alien" stations from accessing the medium.*

As reported in Section 3.3.2, the use of pseudo IEEE 802.15.4 signals by the PNS to synthesize interference is effective in blocking standard IEEE 802.11 transmissions. Provided that these signals have enough bandwidth and power to hinder IEEE 802.11 communications, it is plausible to assume that they are effective in hindering "alien" stations based on the IEEE 802.15.4 standard, operating in an overlapping channel. This assumption was experimentally validated in Section 3.5.2.

Finally, it is assumed that the propagation delays can be negligible when compared to the length of the CCA or IFS, considering low power communication technologies such as IEEE 802.15.4. Provided that the nominal indoor range of this technology is typically 10 meters, the associated propagation delay is approximately 33 nanoseconds, which is much smaller than the corresponding CCA (10 microseconds) or IFS (192 microseconds).

IEEE 802.11 Interference

The first addressed interference scenario corresponds to the IEEE 802.11 protocol. The motivation for evaluating the WFTT protocol timing in open environments encompassing stations employing this technology spans from its wide adoption to support wireless local area networks. Because the IEEE 802.11 technology can be found in a broad range of domains (domestic, industrial, medical, etc.), it is a strong candidate to pose limitations to other co-located technologies operating in the same region of the spectrum, as discussed in Section 2.3.3. In this scenario, as documented in Figure 4.8, the medium will become unprotected during the intervals where no interference (protective or *black-burst*) is being propagated in the medium. This figure is a simplification of Figure 4.6, illustrating explic-

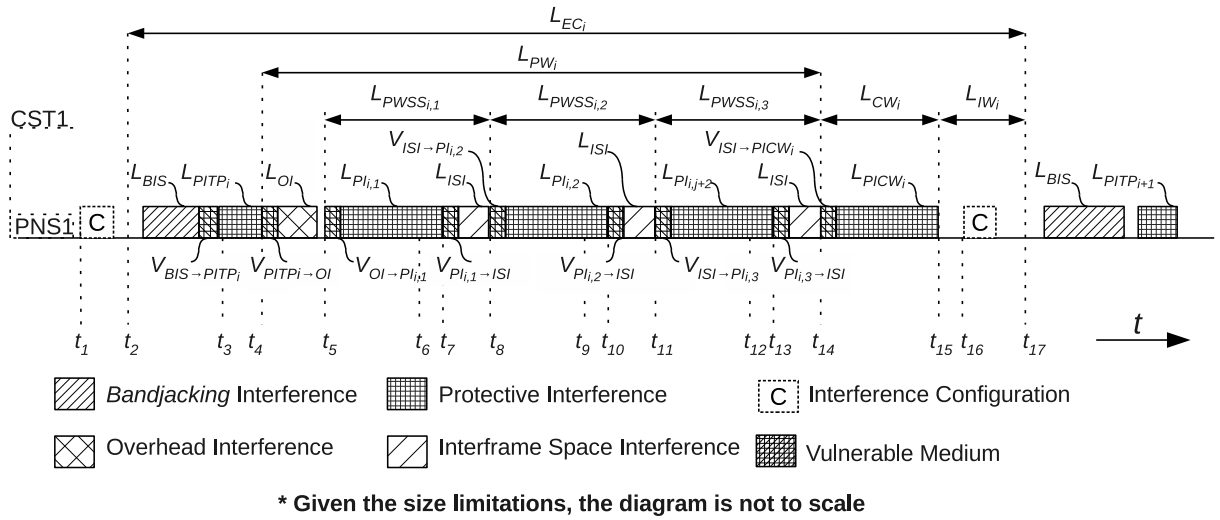


Figure 4.8: WFTT vulnerable intervals to IEEE 802.11 interference

itly the idle medium intervals of time. The medium will become vulnerable (V) after the *bandjacking* sequence in the following periods of time:

- $V_{BIS \rightarrow PITP_i}$ between the end of a *bandjacking* interference sequence (L_{BIS}) and the beginning of a Protective Interference sequence for the TP (L_{PITP_i}), in a given EC i.
- $V_{PITP_i \rightarrow OI}$ between the end of the protective interference sequence for the TP (L_{PITP_i}) and the beginning of the overhead interference sequence (L_{OI}), in a given EC i.
- $V_{OI \rightarrow PI_{i,1}}$ between the end of the overhead interference sequence (L_{OI}) and the beginning of the first protective data interference sequence ($L_{PI_{i,1}}$), in a given EC i.
- $V_{PI_{i,j} \rightarrow ISI}$ between the end of a protective data interference sequence (e.g., $L_{PI_{i,j}}$) and the beginning of the subsequent Inter Frame Space Interference sequence (L_{ISI}) in a given Slave Slot (e.g., $L_{PWSS_{i,j}}$).
- $V_{ISI \rightarrow PI_{i,j+1}}$ between the end of an inter-frame space interference sequence (L_{ISI}), in a given slave slot (e.g., $L_{PWSS_{i,j}}$), and the beginning of the subsequent protective data interference sequence (e.g., $L_{PI_{i,j+1}}$) in the following slave slot (e.g., $L_{PWSS_{i,j+1}}$).
- $V_{ISI \rightarrow PICW_i}$ between the end of an inter-frame space interference sequence (L_{ISI}) of the last slave slot and the beginning of the protective interference in the following CW (L_{PICW_i}), in a given EC i.

The integrity of the PW can only be ensured if "alien" stations are not able to initiate a transmission after the end of the BIS and the conclusion of the PW. As such, based

on the presented assumptions and considering that the bandwidth employed in the data transmissions is much smaller than the bandwidth of the "alien" interference ($BW_{data} \ll BW_{alien}$), the maximum period of time in which the medium becomes idle during the aforementioned interval is represented by Equation 4.8.

$$
\begin{aligned}
V_{PW_{BW_{data} \ll BW_{alien}}} = & \\
& max\{V_{BIS \rightarrow PITP_i}, V_{PITP_i \rightarrow OI}, V_{OI \rightarrow PI_{i,1}}, \\
& V_{PI_{i,j} \rightarrow ISI}, V_{ISI \rightarrow PI_{i,j+1}}\} = \\
& max\{D_{IS} + J_{PITP_i}, D_{IS} + J_{OI_i}, D_{IS} + J_{PI_{i,j}} + max(J_{pTP_{i,k}}), \\
& D_{IS} + J_{ISI_{i,j}}\}, \forall i, k \in \mathbb{N}_1 \wedge j \in [1...N_{PPW_i}]
\end{aligned}
\tag{4.8}
$$

The PI generated by the PNS is assumed to make the medium be perceived as busy by IEEE 802.11 "alien" stations. Hence, in order to keep the medium free from these stations' transmissions during the CW, the maximum period of time in which it can be idle is given by Equation 4.9.

$$
\begin{aligned}
V_{CW_{BW_{data} \ll BW_{alien}}} = max\{V_{ISI \rightarrow PICW_i}\} = & \\
max\{D_{IS} + J_{PICW_i}\}, \forall i \in \mathbb{N}_1 &
\end{aligned}
\tag{4.9}
$$

Provided that the IEEE 802.11 "alien" interferers' MAC dictates that a transmission can only start after an idle minimum period of $L_{SIFS_{IEEE802.11}} = 10$ microseconds (see Table 4.4), the integrity of the PW and of the CW is kept if the results derived from Equations 4.8 and 4.9 are bellow this bound. As Equation 4.8 indicates, the maximum vulnerable period of time in the PW window depends of both the PNS response time and of the processing capabilities of the slaves. Conversely, Equation 4.9 suggests that the medium vulnerability in the CW is only dependent of the PNS implementation.

Table 4.4: IEEE 802.11 CSMA parameters (2.4 GHz band)

PHY Layer	$L_{SIFS_{MIN}}$	$L_{CCA_{MIN}}$
FHSS	28 μs	27 μs
DSSS	10 μs	$\leq$ 15 μs
HR/DSSS	10 μs	$\leq$ 15 μs
Long Slot ERP-OFDM	10 μs	$\leq$ 15 μs
Short Slot ERP-OFDM	10 μs	$\leq$ 15 μs
MIMO	10 μs	$<$ 15 μs

As evaluated in Section 3.4.2, the current PNS implementation presents limitations concerning its response time. It was observed that the minimum and maximum latency to turn on interference on a given channel ranges from 124 to 199 microseconds. The turnoff latency ranges from 74 to 108 microseconds. These values are affected by a measurement offset introduced by the BeeMon monitor that varies from 65 microseconds to 81 microseconds. The BeeMon monitor implementation and evaluation are detailed in Appendix D. Assuming that the minimum turn on/off latencies were affected by the minimum BeeMon offset and, conversely, that the maximum BeeMon offset affected the maximum interference turn on/off latencies, it is possible to obtain an indicative range of latencies for the current PNS implementation timeliness. These values can be expressed by a constant part, which corresponds to the minimum latency, and by an associated jitter, corresponding to the difference between the minimum and the maximum latencies. Therefore, the interference turn on can be represented by a delay of 59 microseconds and a maximum jitter of 59 microseconds while the turnoff can be achieved with a delay of 9 microseconds and a maximum jitter of 18 microseconds. WFTT

The PNS implementation presented and evaluated in Section 3.4 exhibits values of latency and jitter significantly higher than 10 microseconds. The aforementioned results indicate that the interference turn on delays are characterized by a constant delay of $D_{IS} =$ 59 microseconds and a maximum jitter of $J_{PITP_{MAX}} = J_{OI_{MAX}} = J_{PI_{MAX}} = J_{ISI_{MAX}} =$ 59 microseconds. Although substantially lower than the interference turn on delay, the turnoff delay is not negligible, as it was assumed. This PNS responsiveness impairment is motivated by the use of SPI commands to control the state of the transceivers' power amplifiers. Hence, since it is possible to drive directly the enable/disable signal of these amplifiers, it is reasonable to assume that D_{IS}, J_{PITP_i}, J_{OI_i}, $J_{PI_{i,j}}$, $J_{ISI_{i,j}}$ and J_{PICW_i}, represented in Equations 4.8 and 4.9 can significantly reduced to become negligible when compared to the 10 microseconds bound.

Regarding the processing power of the slaves, the $max(J_{pTP_{i,k}})$ parameter value will be negligible if the adopted stations employ a common hardware architecture encompassing a fast processor and execute the same firmware instructions to process the trigger packet. Therefore, this analysis suggests that a WFTT network can be implemented and deployed in an open environment populated by IEEE 802.11 standard stations without risking an impact on its performance due to uncontrolled IEEE 802.11 "alien" transmissions, as long as a PNS with a fast time response is employed to synthesize interference and the adopted slave stations are supported on an adequate shared hardware architecture.

IEEE 802.15.4 Interference

The second interference scenario being addressed corresponds to the IEEE 802.15.4 protocol, which is also one of the most widespread wireless technologies in the market for personal area networks. Provided that it lies at the base of several popular communication protocols such as ZigBee and WirelessHART, this technology can be found in many applications, ranging from domotics to industrial processes, for example. Therefore, it is a potential candidate to contend for the 2.4 GHz band where the WFTT operates. In a scenario where data and "alien" transmissions occupy overlapping spectrum regions and $BW_{data} \approx BW_{alien}$ or $BW_{data} > BW_{alien}$, the medium will be protected only when WFTT transmissions or *black-burst* interference sequences are ongoing, as represented in Figure 4.9. Otherwise, "alien" stations may find the medium idle, even if the master's PNS is synthesizing protective interference. Figure 4.9 is a simplification of Figure 4.6, illustrating explicitly the intervals of time where the medium becomes idle and can be compromised by "alien" transmissions. The following list summarizes these intervals:

- $V_{BIS \rightarrow TP_i}$ between the end of a *bandjacking* interference sequence (L_{BIS}) and the beginning of a TP (L_{TP_i}), in a given EC *i*.
- $V_{TP_i \rightarrow OI}$ between the end of the protective interference sequence for the TP (L_{TP_i}) and the beginning of the overhead interference sequence (L_{OI}), in a given EC *i*.
- $V_{OI \rightarrow PD_{i,1,k}}$ between the end of the overhead interference sequence (L_{OI}) and the beginning of the first protected data transmission ($L_{PD_{i,1,k}}$) performed by the k^{th} slave station in a given EC *i*.
- $V_{PD_{i,j,k} \rightarrow ISI}$ between the end of a protected data transmission (e.g., $L_{PD_{i,j,k}}$), performed by the k^{th} slave station in a given j^{th} slave slot, and the beginning of the subsequent Inter Frame Space Interference sequence (L_{ISI}).
- $V_{ISI \rightarrow PD_{i,j,k}}$ between the end of an inter-frame space interference sequence (L_{ISI}) and the beginning of the subsequent protective data transmission (e.g., $L_{PD_{i,j,k}}$) by the k^{th} slave station in the j^{th} slave slot (e.g., $L_{PWSS_{i,j}}$).

As in the IEEE 802.11 case, the integrity of the PW can only be ensured if "alien" transmissions are not possible to occur between the end of the BIS and the end of the PW. Equation 4.10 provides an expression that allows to determine the maximum period of time during which the medium is idle between the end of the BIS and of the PW.

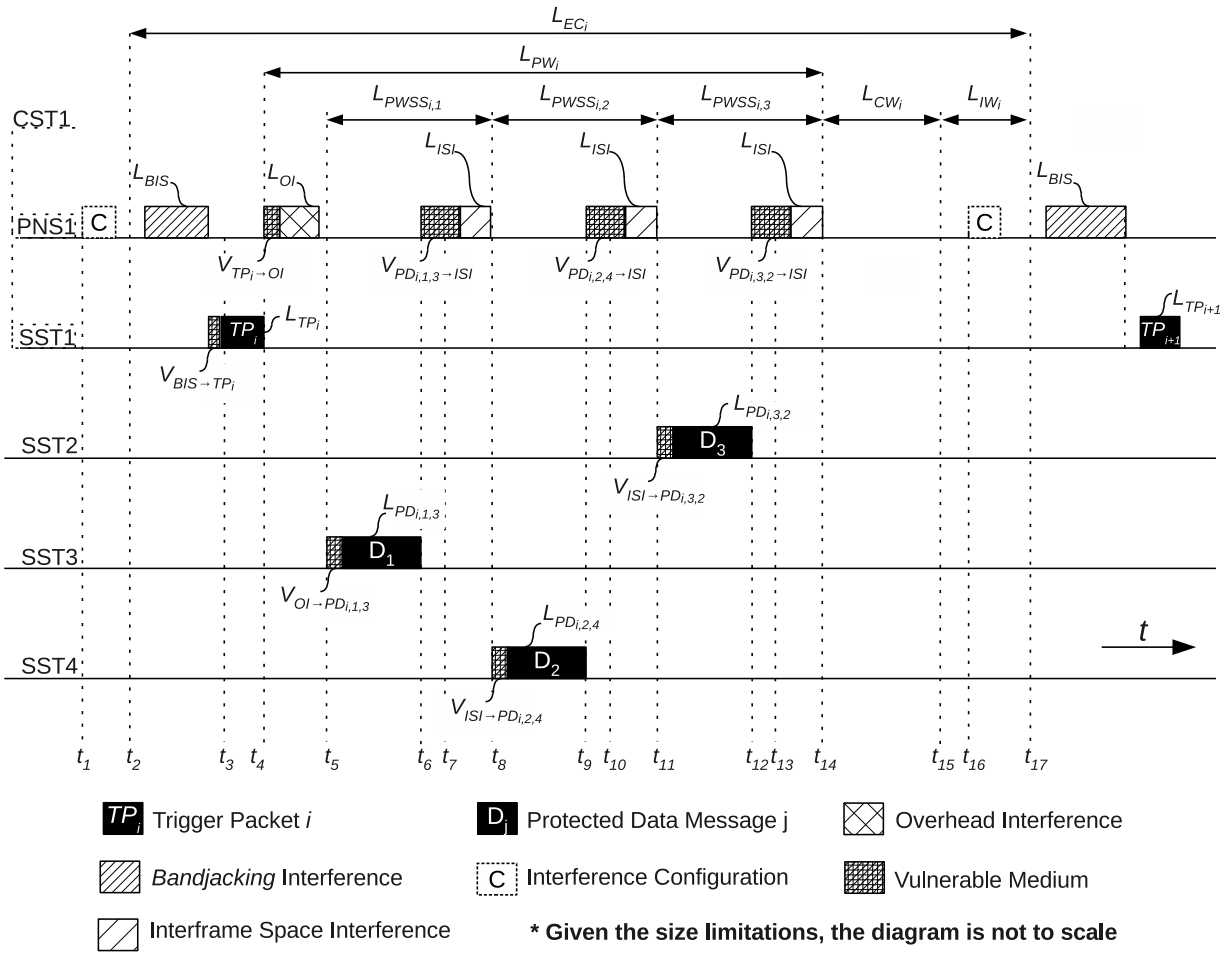


Figure 4.9: WFTT vulnerable intervals to IEEE 802.15.4 interference

$$
\begin{aligned}
&V_{PW_{BW_{data} \geq BW_{alien}}} = \\
&\quad max\{V_{BIS \to TP_i}, V_{TP_i \to OI}, V_{OI \to PD_{i,1,k}}, V_{PD_{i,j,k} \to ISI}, V_{ISI \to PD_{i,j,k}}\} = \\
&\quad max\{J_{TP_i}, D_{IS} + J_{OI_i}, max(J_{pTP_{i,k}}) + J_{PD_{i,j,k}}, J_{PD_{MAX}} - J_{PD_{i,j,k}} + D_{IS} + J_{ISI_{i,j}}, J_{PD_{i,j,k}}\}, \\
&\quad \forall i, k \in \mathbb{N}_1 \wedge j \in [1...N_{PPW_i}]
\end{aligned}
\tag{4.10}
$$

where:

$$J_{PD_{MAX}} = max(J_{PD_{i,j,k}})$$

Regarding the CW, because the protective interference has no impact on hindering "alien" stations that have a bandwidth similar to or narrower than the one of the WFTT

network, it is not possible to avoid contention among both types of stations. In other words, the WFTT CW transmissions will not be protected against contention from "alien" stations employing a similar technology.

Provided an environment populated with IEEE 802.15.4 interferers, "alien" transmissions can only be initiated after a period of time corresponding to the clear channel assessment (CCA) of this technology, which is given by $L_{CCA_{IEEE802.15.4}} = 128$ microseconds (Table 4.5). Hence, the PW integrity in this environment is kept only if $V_{PW_{BW_{data} \geq BW_{alien}}} < 128$ microseconds. As in the IEEE 802.11 interference scenario, D_{IS}, J_{OI_i} and $J_{ISI_{i,j}}$ are dependent of the response time of the PNS and, according to the premises laid for that scenario, they will be much smaller than the 128 microseconds bound.

Table 4.5: IEEE 802.15.4 CSMA parameters (2.4 GHz band)

PHY Layer	$L_{SIFS_{MIN}}$	$L_{CCA_{MIN}}$
DSSS	192 μs	128 μs

On the other hand, J_{TP_i}, $J_{pTP_{i,k}}$, $J_{PD_{i,j,k}}$ and $J_{PD_{MAX}}$ depend on the response time of standard stations. To simplify the analysis, assume, as before, that the slave stations share an hardware architecture encompassing a fast processor, which executes a common set of instructions to process the trigger packet, resulting in a negligible $J_{pTP_{i,k}}$. If the jitter of initiating a TP (J_{TP_i}) is smaller than the maximum jitter of initiating a protected transmission ($J_{PD_{MAX}}$), Equation 4.10 can be further simplified to Equation 4.11. Otherwise, the vulnerability to "alien" transmissions is only dependent of J_{TP_i}.

$$V_{PW_{BW_{data} \geq BW_{alien}}} = J_{PD_{MAX}} + D_{IS} + J_{ISI_{i,j}}, \forall i \in \mathbb{N}_1 \wedge j \in [1...N_{PPW_i}] \tag{4.11}$$

Provided the 128 microseconds bound for an idle interval and recalling the previously obtained D_{IS} and $J_{ISI_{MAX}}$ values (59 microseconds each), only a slack of 10 microseconds is left to cope with the jitter of transmitting a protected packet, which will be challenging to meet in practice. Therefore, an improvement of the PNS response time is required to lower the delay and jitter associated to the transmission of interference (D_{IS} and $J_{ISI_{i,j}}$, respectively). This will provide a larger margin for the protect data transmission jitter ($J_{PD_{i,j,k}}$). As described in the IEEE 802.11 interference scenario, the solution corresponds to enabling the direct driving of the PNS power amplifiers, thus significantly reducing

the latency of switching the interference state. With this in mind, the presented analysis points towards a WFTT network that can be implemented and deployed in an open environment populated by IEEE 802.15.4 standard stations. Although the network is vulnerable to contention from "alien" stations during the CW, the consistency of the real-time communications performed in the PW is ensured by the protocol's operation.

4.3.2 Timeliness

The timeliness of the WFTT protocol is determinant for its adoption in real-time applications. Besides the reliability of the transmissions, the transmission latency and jitter can impair its usage in demanding real-time applications. Along these lines, this subsection presents a study of the WFTT protocol timeliness focused on the transmission latency and jitter for both types of traffic: synchronous and asynchronous. The transmission delay definition corresponds to the interval of time elapsing between the instant where a transmission is ordered to the transceiver, after the packet was successfully loaded into the transceiver's transmission buffer, and the instant of the packet detection at the receiver, marked by an interrupt event at the transceiver. The transmission jitter is defined as the difference between the minimum and maximum transmission latencies.

Synchronous Traffic

The synchronous traffic communicated in a WFTT EC can be divided in two types: control and data. The first corresponds to the data carried by the trigger packet to inform the slaves about the structure of the EC and of the scheduled protected transmissions. This traffic is generated with a period given by the EC length and has a variable duration, according to the number of protected scheduled transmissions. The transmitter delay and jitter of this type of traffic in a given EC i is represented by TD_{TP_i} and TJ_{TP_i}, respectively. As shown in Equation 4.12, TD_{TP_i} encompasses the delay to initiate a TP transmission (D_{TP}), the actual length of the transmission (L_{TP_i}) and the delay to detect its transmission at the slaves (D_{dTP}). The latter, for simplification purposes and compared to the length of the TP (L_{TP_i}), is assumed to be negligible. Equation 4.13 defines that the transmission jitter is the same of initiating the TP transmission.

$$TD_{TP_i} = D_{TP} + L_{TP_i} + D_{dTP}, \forall i \in \mathbb{N}_1 \tag{4.12}$$

$$TJ_{TP_i} = J_{TP_i} \forall i \in \mathbb{N}_1 \tag{4.13}$$

An estimate for both delay and jitter can be derived by considering a typified scenario and making some basic assumptions. Consider a WFTT network encompassing three real-time stations performing one protected transmission each per PW, for example. Provided that the TP must contain the scheduling information of these three transmissions, the overall length of the TP is 48 bytes, including the IEEE 802.15.4 synchronization, PHY and MAC headers with the lengths of 5, 1 and 11 bytes, respectively, besides the MAC footer with a length of 2 bytes. Therefore, the theoretical length of the TP transmission (L_{TP_i}) is of 1536 microseconds.

In Section 3.4.2, it was observed that the Microchip IEEE 802.15.4 MRF24J40 transceiver is affected by a delay of 555 microseconds and a jitter of 36 microseconds to initiate a transmission, i.e., between the instant when the SPI packet transmission order is issued (of a packet previously loaded into the transmission buffer) and the instant when the associated transmission is effectively initiated. Although these values were measured placing the MRF24J40 transceiver in a special mode of operation called *turbo mode*, it is reasonable to assume that this transceiver will exhibit a similar latency when instructed to perform standard IEEE 802.15.4 data transmissions. Thus, it is acceptable to consider that the delay (D_{TP}) and jitter (J_{TP_i}) to initiate the TP can be represented by the values of 555 and 36 microseconds, respectively.

Regarding the delay to detect the trigger packet at the WFTT slave stations (D_{dTP}), the latency to trigger the external interrupt line at the transceiver and to attend the interrupt service routine at the standard station can be assumed negligible when compared to the length of the packet. In this sense, substituting the presented values in Equations 4.12 and 4.13, it is possible to estimate a TP transmission delay of 2091 microseconds and a jitter of 36 microseconds.

The synchronous data traffic mentioned before corresponds to the real-time information exchanged between slave stations in the PW of the WFTT's EC. The transmission period of this type of traffic depends of the schedule contained in the TP, i.e., it is defined in a per EC basis. Equations 4.14 and 4.15 represent the transmission delay and jitter associated to this type of traffic, considering a real-time transmission performed by the slave station k in the slave slot j of the i^{th} EC.

$$TD_{PD_{i,j}} = D_{PD} + L_{PD_{i,j}} + D_{dPD}, \forall i \in \mathbb{N}_1 \wedge j \in [1...N_{PPW_i}] \tag{4.14}$$

$$TJ_{PD_{i,j,k}} = J_{PD_{i,j,k}} \forall i,k \in \mathbb{N}_1 \wedge j \in [1...N_{PPW_i}] \tag{4.15}$$

As in the TP timeliness characterization, it is possible to estimate the transmission delay and jitter by establishing some basic assumptions regarding the parameters of Equations 4.14 and 4.15. Considering that a WFTT real-time data packet has a length of 29 bytes, its transmission length ($L_{TP_{i,j}}$) occupies a time window of 928 microseconds. Assuming, as before, that D_{dPD} is negligible and the values of delay (555 microseconds) and jitter (36 microseconds) presented in Section 3.4.2 for initiating a packet transmission, it is possible to obtain a real-time packet transmission delay and jitter estimate of 1483 and 36 microseconds, respectively.

Both types of synchronous traffic estimates could be improved by effectively measuring the activation delay and jitter associated to the transmission of standard IEEE 802.15.4 packets and the latency of detecting a packet reception.

Asynchronous Traffic

The asynchronous traffic can be triggered at any instant of the contention window as long as it can be guaranteed that its completion is concluded within the timing bounds of this window. The overall transmission delay associated to WFTT contention transmissions is defined by Equation 4.16. As documented, it encompasses the delay elapsing between the transmission trigger and the instant in which the transceiver can begin the effective propagation of the transmission (D_{CD}); the length of the effective contention transmission (L_{CD}); and, finally, the delay to detect the contention data packet at the receiver (D_{dCD}).

$$TD_{CD} = D_{CD} + L_{CD} + D_{dCD} \tag{4.16}$$

Equation 4.17 presents the transmission jitter of the m^{th} contention data packet transmission. This jitter comprises two components, one related to the time window elapsing between the transmission order and its effectiveness in the medium (J_{CD_m}), and the other to the backoff algorithm of the CSMA-CA mechanism employed in standard IEEE 802.15.4 communications (J_{BK_m}).

$$TJ_{CD_m} = J_{CD_m} + J_{BK_m} \forall m \in \mathbb{N}_1 \tag{4.17}$$

As before, assuming that the delay and jitter of initiating a transmission can be approximately represented by the values observed in Section 3.4.2, the D_{CD} and J_{CD_m} parameters can take the values of 555 and 36 microseconds. The overall length of a data contention packet is 25 bytes. Therefore, its transmission has an effective duration of 800 microseconds. By assuming that D_{dCD} is negligible, it is possible to obtain a representative estimate of

the contention packet transmission delay using Equation 4.16. Thus, its value corresponds to 1355 microseconds.

Regarding the jitter expressed in Equation 4.17, the missing parameter is the jitter introduced by the IEEE 802.15.4 backoff algorithm. Recalling Section 2.1.2, in the unslotted version of the CSMA/CA, before initiating the CCA, a random number of backoff periods ranging from 0 to $2^{BE} - 1$ must be enforced. Provided that each backoff period has a length of 20 symbols, the symbol rate is 62500 symbols *per* second and the typical backoff exponent used in broadcast packets is 3, the maximum delay experienced during the backoff phase is bounded by a value of 2240 microseconds ($J_{BK_{MAX}} = 2240$). Therefore, an estimate of 2276 microseconds can be obtained for the maximum jitter of a contention data transmission.

As in the synchronous traffic case, this estimate of the delay and jitter can be improved by measuring the effective activation delay and jitter associated to the transmission of standard IEEE 802.15.4 packets. Moreover, the latency of detecting a packet reception could be evaluated in order to provided a more accurate timeliness estimate.

4.4 Summary

This chapter presented the foundations, operation and characterization of the Wireless Flexible Time-Triggered (WFTT) protocol. It provided a short review of the protocol upon which it was inspired (FTT), describing its architecture, operation and distinctive features. Afterwards, the specification of the WFTT protocol was presented focusing on its architecture and operation. The key elements that characterize the protocol's behavior (elementary cycle, packet structure, temporal isolation, etc.) were introduced and analyzed. Finally, in order to obtain an improved characterization of the WFTT protocol, a study focused on its implementation constraints and timeliness was presented. This study was conducted on an explicative scenario encompassing WFTT stations using the IEEE 802.15.4 protocol and "alien" stations attempting to perform transmissions on a common region of the spectrum using one of two communication technologies: IEEE 802.15.4 and IEEE 802.11.

Regarding the implementation feasibility, it was found that in a scenario employing the IEEE 802.15.4 technology to perform data communications in an environment affected by "alien" IEEE 802.11 noise ($BW_{data} \ll BW_{alien}$), the maximum period of time during which the medium can be idle without risking being compromised is 10 microseconds. Likewise,

the maximum idle time interval between the end of the *black-burst* interference sequence initiating an EC and the end of its PW is expressed by $V_{PW_{BW_{data} \ll BW_{alien}}}$. Therefore, in order not to compromise the PW, this parameter must be kept smaller than 10 microseconds. Using the current WFTT PNS implementation this is not possible due to the latency and jitter introduced by the use of SPI commands to turn on/off the transmission of interference. This limitation can be overcome by driving directly the PNS transceivers' power amplifiers.

More on the implementation feasibility. In a scenario where both WFTT and "alien" stations employ the IEEE 802.15.4 technology ($BW_{data} \geq BW_{alien}$), the maximum interval of time during which the medium becomes unprotected in the interval elapsing from the *black-burst* interference sequence to the end of the PW is given by the $V_{PW_{BW_{data} \geq BW_{alien}}}$ parameter. As demonstrated, in order to maintain the consistency of the PW, this parameter must be bounded by a value of 128 microseconds, which corresponds to the duration of the IEEE 802.15.4 CCA operation. Although the current PNS implementation provides some margin for the slave stations to theoretically meet this bound, its implementation using resource limited embedded devices seems challenging. Hence, as in the former scenario, an improvement of the PNS implementation to provide a larger slack in the timeliness of the slave stations is required.

The timeliness of the WFTT protocol was analyzed focusing on both the synchronous and asynchronous traffic. In the first case, the delay and jitter associated to both trigger and real-time packet transmissions was studied. Besides presenting the expressions ruling the delay and jitter experienced by these two packet types, an estimate based on reasonable assumptions is provided. The delay and maximum jitter of the trigger packet transmission is given by the $TD_{TP_i} = 2091$ microseconds and $TJ_{TP_i} = 36$ microseconds estimates, respectively. The components that mostly contribute to the TP transmission delay are the activation delay and the duration of the effective transmission. Concerning the timeliness estimates for the real-time packets, $TD_{PD_{i,j}} = 1483$ microseconds and $TJ_{PD_{i,j,k}} = 36$ microseconds correspond to the transmission delay and maximum jitter that these packets may suffer. As in the former case, the transmission activation delay and its length are the most representative latency components.

The asynchronous traffic timeliness was also studied in this chapter. Provided that all data transmissions in the WFTT protocol are performed using broadcasts, this study focused on the broadcast of packets according to the IEEE 802.15.4 CSMA/CA unslotted scheme. The result of this study was the identification of two expressions that represent the

Table 4.6: WFTT transmission timeliness estimates

Type	Delay (μs)	Jitter (μs)
Trigger Packet	2091	36
Real-time Packet	1483	36
Contention Packet	1355	2276

delay and jitter of a contention based transmission using the IEEE 802.15.4 protocol. Besides performing a timing analysis and identifying the key delay components contributing to the timeliness parameters, this chapter also provides the corresponding estimate using plausible assumptions. In this sense, it was concluded that the transmission delay and jitter of broadcast contention data packets is of 1355 and 2276 microseconds, respectively. In this case, besides the activation delay and length of the transmission, the most representative delay component of the packet's timeliness is the jitter associated to the IEEE 802.15.4 CSMA/CA backoff procedure. The timeliness estimates of the trigger, real-time and contention packets are summarized in Table 4.6 for easier reference.

"No amount of experimentation can ever prove me right; a single experiment can prove me wrong."

Albert Einstein (1879 - 1955)

5 Framework Implementation

The previous chapter presented and analytically validated the WFTT protocol. In this chapter, an instantiation of the protocol is provided using practical devices. A description of the envisaged operation and of the implementation of master and slave devices is also discussed. This description is focused on the device's individual architecture and operation.

5.1 Envisaged WFTT Operation

The instantiation of the WFTT protocol using real devices must account for the limitations associated to an experimental implementation, namely the processing delays of handling events. It is important that these delays are acknowledged and well characterized in an early stage of the implementation since they pose timing restrictions that, potentially, may cause an incorrect behavior of the stations participating in the WFTT network and are only realized in the later states of development (e.g., during the tests). It is also important to have a global perspective of the WFTT network operation, including all the expected events that may occur in a single elementary cycle as well as the different types of devices participating in the network. Besides allowing a better understanding of the overall operation, it makes it possible to identify the interactions and dependencies among different stations and elements.

Figure 5.1 depicts the WFTT operation of the envisaged implementation during the length of one elementary cycle. As illustrated, three stations are represented: one master and two slaves. While the slave stations are only able to transmit data packets, the master station is also capable of performing protective (1) and *black-burst* (2) interference transmissions using the PNS. The (1) notation refers to the (1) element marked in Figure 5.1 and it will be used in the remaining of this chapter to unequivocally mention other

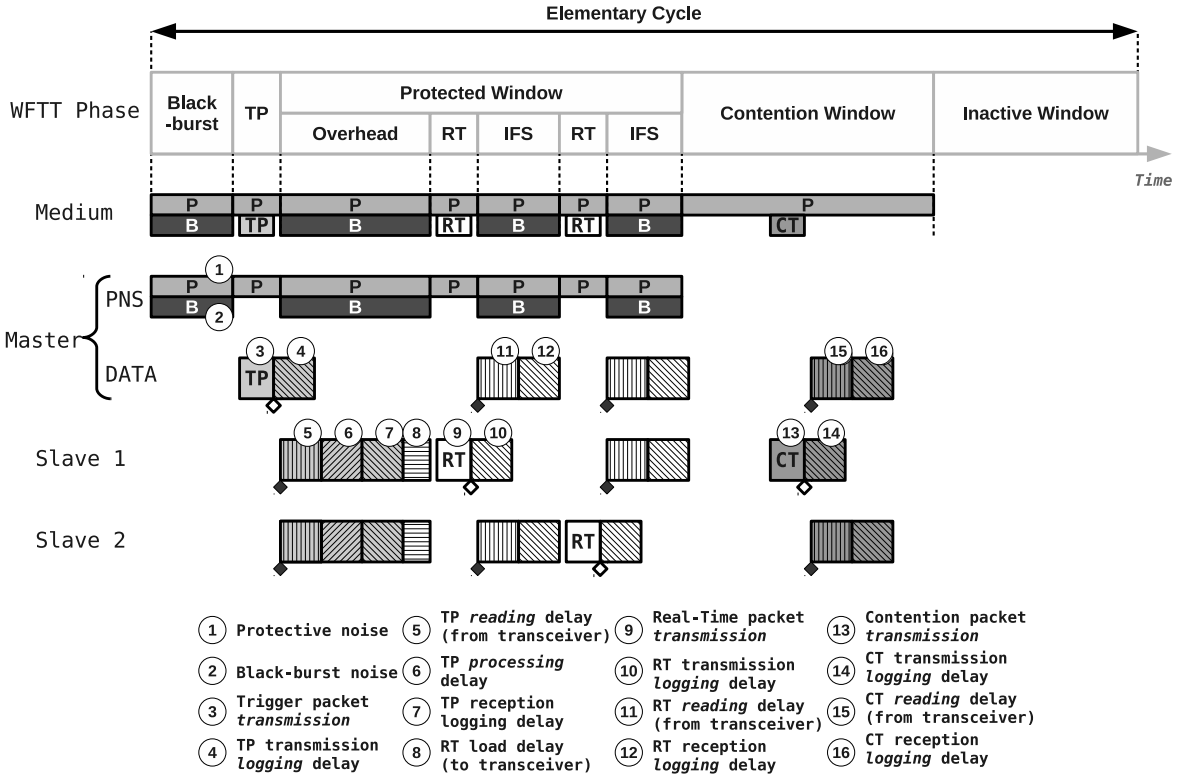


Figure 5.1: WFTT elementary cycle implementation

relevant elements of this figure.

As represented, the *black-burst* noise is transmitted in the same IEEE 802.15.4 channel as the data packets. The protective noise is issued on a lateral channel, which does not overlap the data channel. Figure 5.1 documents these three types of transmissions in two timelines: one associated to the *black-burst* noise (B) and data packets; and the other to the protective interference (P). In order to facilitate the association between the station which performs a given transmission and the corresponding medium occupancy, all transmissions are represented simultaneously on the timeline of the issuing station and on the timeline of the medium.

As defined in the previous chapter, the elementary cycle encompasses the protected, contention and inactive windows. The protected window consists of time slots, which are previously reserved for priority communications between real-time stations. The transmissions performed in the contention window use the CSMA/CA mechanism to contend for the medium. When appropriate, the PNS synthesizes protective noise, *black-burst* noise or both, depending of the phase of the WFTT elementary cycle.

The envisaged WFTT operation relies on a tight elementary cycle control by the master

station and a configurable operation by the slave stations. As to the former, using the PNS, the master synthesizes noise in both *black-burst* and protective channels during a period of time corresponding to the maximum length of any packet being transmitted by an "alien" station. In this case, on account of the conclusions of Section 2.3, Wi-Fi was considered the "alien" technology with the highest potential impact on the IEEE 802.15.4 network operation. After the initial noise sequence, "alien" stations are (eventually) prevented from transmitting on the medium and the master station transmits immediately after the trigger packet ③, which encodes the schedule of the current elementary cycle. This trigger packet (TP) includes the information described in Figure 4.5, which, among other parameters, defines both the number of real-time slots (and offsets) in the protected window and the durations of the contention window and of the elementary cycle. The TP is accompanied by protective noise, which hinders Wi-Fi stations from finding the medium idle and initiating the transmission of packets.

As documented in Figure 5.1, in the envisaged implementation, besides the TP, two additional types of data transmissions will occur during the length of an elementary cycle: real-time ⑨ and contention ⑬ transmissions. The support of real-time and contention transmissions aims at meeting the requirements of the WFTT protocol for issuing packets in the protective and contention windows, respectively. The occurrence of these two types of transmissions by the same slave in a elementary cycle is not common in real world deployments. Usually a slave has communication requirements fitting on either the WFTT protected or contention windows. However, to provide a more complete vision of the WFTT operation, two types of transmissions are explained in a single elementary cycle. Furthermore, the firmware implementation of the slave stations was conducted in order to enable switching their transmission pattern in runtime, i.e., modifying the station type and, hence, change the type of packets sent by that station (real-time or contention). By allowing changing the type of packets being transmitted using external commands in runtime, it is possible to easily devise an automated control system to monitor and evaluate the WFTT protocol operation. This is of paramount importance, given that it can significantly reduce the amount of time spent preparing the WFTT testbed and performing the associated trials. In the remaining of this section, a detailed explanation of the envisaged WFTT protocol operation is presented with an emphasis on the delays introduced by the limitations of a practical implementation.

The WFTT example presented in Figure 5.1 shows a protected window including an overhead delay and two real-time transmission slots. After the reception of the trigger

packet (3), all real-time stations take some time to read it from their transceivers (5), process it (6), log it (7) and, finally, load a real-time packet (8) to the transceiver for transmission. The logging action enables an external characterization of the WFTT timeliness. In this sense, a guard interval is reserved to enable an (external) assessment system saving and time-stamping the received packet. The WITAS is the adopted tool for this purpose. Details about its architecture and operation are available in Appendix A.

The timeliness valuation of the protocol requires the support of specific features on the WFTT stations. Hence, an early assessment of the impact of this features in the WFTT implementation is required since, otherwise, a later inclusion may result in modifications whose complexity can disrupt the timeliness of the WFTT protocol already implemented. In this sense, the implementation of the WFTT defines that whenever an event occurs (transmission/reception of a packet), it is logged according to a predefined process whose delay is foreseeable and can be considered in the implementation. The accounting of these delays occurs for the trigger packet (4), real-time packet (10) and the packets transmitted in the contention (14) window, as documented in Figure 5.1. The propagation delay is considered negligible when compared to the delays mentioned above for a wireless local area network. The scheduled real-time offsets encoded in the TP account for the sum of these delays, guaranteeing that the slaves are ready to transmit their packets when the master is expecting them to be transmitted.

During the period in which the slave stations are processing the TP, the master station occupies the medium to guarantee its inactivity when the first real-time station starts its transmission in the assigned slot. Hence, it drives the PNS to synthesize both protective (1) and *black-burst* (2) interference during this interval. Then, at the offset instant encoded in the TP, the slave station scheduled for transmitting in the first slot triggers the corresponding real-time packet. The master station protects this transmission by ensuring that the PNS generates protective interference (1) during the length of the packet. Afterwards, the slave station logs (10) the transmitted real-time packet while the receiving stations (master and slaves) read it (11) from their transceivers and proceed to its logging (12). During this interval, the medium is protected from Wi-Fi transmissions by both protective (1) and *black-burst* (2) interference. The cycle described for the first real-time packet transmitted in the protected window is replicated for the second real-time transmission, as shown in Figure 5.1.

The WFTT protocol defines that there must be an IFS period between consecutive data transmissions to ensure that, for example, slave stations have enough time to perform their

transmissions and also to receive the transmissions from neighbor slave stations. However, it is important to notice that a more agile mechanism could be devised in the protected window, if slave transmissions targeted exclusively the master. In this case, the IFS could be significantly reduced, since the master remains in the receive state during the whole duration of the protected window.

As introduced, the firmware implemented for the slave stations was designed to allow an on-line switching of the type of packets being transmitted. In the description of the protocol presented above it was assumed that the slave stations were configured to transmit real-time packets according to the schedule defined by the TP. In order to explain the master and slave operation on the contention window, it is assumed that slaves are specifically configured to transmit packets in this window. In this sense, they listen to the medium for the TP and, upon its reception, they determine the offset of the contention window beginning using the information carried in the TP. Only one transmission is scheduled per elementary cycle. Its offset follows a uniform random distribution, which is chosen to occur within the window temporal bounds. This offset is added to the window offset defined by the TP. At the scheduled instant, the CT packet is transmitted (13) and logged accordingly (14) by the slave. As before, all stations receiving the packet must read it (15) from their transceivers and log it (16) as well.

As discussed in the previous chapter, the inclusion of the inactive window in the WFTT elementary cycle was motivated by the requirement of allowing other technologies (in this case Wi-Fi or IEEE 802.15.4) to contend for the medium and to be serviced in a periodic fashion. Otherwise, such "alien" technologies would suffer from "starvation" before being able to access the medium.

5.2 Master Station

As aforementioned, the master station is responsible for coordinating the real-time network operation and bridging information from/to the outside world. In this section, the base architecture of the master station is scrutinized both in terms of the employed hardware and of the devised software.

5.2.1 Architecture

A key element of the WFTT hardware design is its centralized nature, in which a single master station should be able to manage a large number of slave stations (several

tens). Therefore, the pressure for cost reduction is shifted towards slave stations, i.e., the implementation of the master station does not have a tight limitation concerning the low-cost requirement. Furthermore, although in some scenarios it could be of interest to support battery powered master stations, in general, it is assumed that they can get energy from the mains supply through a voltage converter. Also, since the master station will operate as a gateway between the low-power wireless network and the outside world, different communication technologies can be selected for this task. Thus, it seems beneficial that these technologies can be enabled in the master station in a flexible way. For example, using a modular approach where each technology is implemented in a module that can be easily attached/detached from the master station board. Further requirements regarding the master station architecture are the possibility of connecting it directly to a computer platform (embedded or not) and the availability of a simple human interface to monitor and change the master station's operation.

Hardware

Figure 5.2 shows a block diagram of the master station. As documented, it encompasses several components: Micro-Controller Unit (MCU), IEEE 802.15.4 *communication module*, *Expansion Port*, *Power Control*, and *Human Interface*. Optionally, a USB interface, *Temperature Sensor* and a battery (*Li-ion*) can also be integrated. The MCU will hold the application responsible for coordinating the real-time wireless network as well as to bridge any communications to the "world outside". Therefore, the MCU not only has to be fast enough to run these applications in a timely fashion, but must also have enough memory to store and run the application and any communication stacks being employed. The adopted MCU is the Microchip's dsPIC33FJ256MC710, a high-Performance 16-Bit Digital Signal Controller (DSC) capable of running at 40 MIPS and providing a flash memory of 256 kB plus a Random-Access Memory (RAM) memory of 30 kB. Besides speed and memory capacity, this DSC presents other features that make it especially adequate for implementing the master station, namely:

- The support of up to nine 16-bit timers, which can paired up to form four 32-bit timers, thus enabling the development of time sensitive applications;

- An interrupt controller with a 5-cycle latency capable of handling up to 67 distinct interrupt events, allowing a low delay/jitter exception handling support;

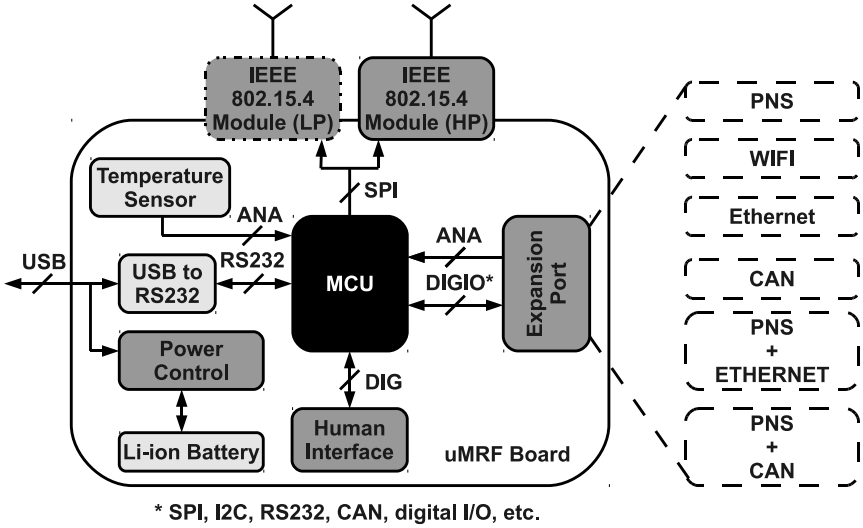


Figure 5.2: Block diagram of the master prototype

- The integration of a Direct Memory Access (DMA) module with eight hardware channels and a 2 Kbytes dual ported DMA buffer area, enabling data transfers between the RAM and the peripherals while the Central Processing Unit (CPU) is executing code, i.e., there is no cycle stealing from the CPU when data transfers occur between DMA-enabled peripherals and the RAM;

- Multiple management features such as, for example, Power-up Timer, Watchdog Timer and Fail-Safe Clock Monitor, which provide an increased assurance of the MCU's correct behavior;

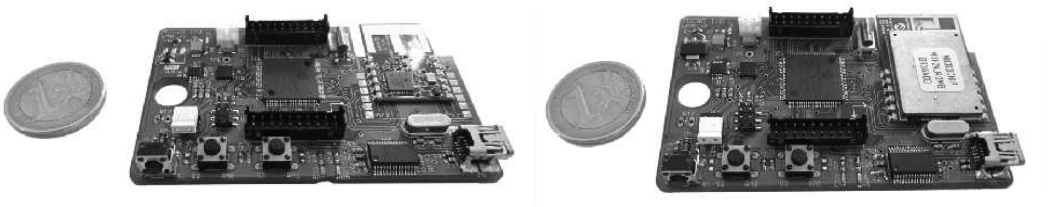
- Several communications interfaces. The dsPIC33FJ256MC710 DSC integrates two Serial Peripheral Interface (SPI) modules, two Inter-Integrated Circuit (I^2C) modules, two Universal Asynchronous Receiver/Transmitter (UART) modules and two Controller Area Network (CAN) modules, allowing the support of a wide variety of peripherals (e.g., memories, communication modules, etc.).

In addition to these features, this DSC encompasses an eight channel Pulse Width Modulation (PWM) module and a quadrature encoder interface module targeting motor control applications as well as a 12-bit, 500 kbps Analog-to-Digital Converter (ADC) with 32 channels capable of sampling two, four or eight inputs simultaneously, targeting sensing applications.

The IEEE 802.15.4 *module* enables short-range low-power wireless communications in the 2.4 GHz Industrial, Scientific and Medical (ISM) band according to the IEEE 802.15.4

standard [74]. The MRF24J40 transceiver was selected in Section 2.4 as the best solution for supporting IEEE 802.15.4 compliant communications. Hence, the master station can mount two IEEE 802.15.4 modules based on this transceiver, which are characterized by different communication ranges: Microchip's MRF24J40MA and MRF24J40MB modules. The first has a typical output power of +0 dBm (36 dB transmit power control range) and a −94 dBm typical sensitivity. The second supports an output power of +20 dBm (56 dB transmit power control range) and a −102 dBm typical sensitivity. Figure 5.3 presents a prototype of the master station with either the MRF24J40MA (*Low-Power*) or the MRF24J40MB (*High-Power*) module assembled. Both modules were enabled in the master station, although mutually exclusively, because of the heterogeneous range requirement posed by different applications. Several reasons accounted for this option, namely the familiarity with the Microchip tool-chain; the easy integration with the adopted MCU; the base transceiver low-cost and low-power operation; the IEEE 802.15.4 standard compliance; and the features provided by the transceiver, highlighted by the hardware support for the CSMA-CA mechanism; the automatic ACK response; the support of the full range of CCA modes and RSSI/LQI; the hardware security engine (AES-128) with support for the CTR, CCM and CBC-MAC modes; and the encryption and decryption support for the MAC and higher layers. The other technical aspects that drove this option were the low-power consumption, as documented in Table 5.1; the simple 4-Wire SPI interface; the integrated oscillator circuitry; the small form factor; and the support for the ZigBee, MiWi, MiWi P2P stacks and proprietary wireless networking protocols, especially those requiring the disabling of the CSMA-CA medium access mechanism. Besides these features, this module encompasses specific hardware allowing both the standard IEEE 802.15.4 transmission scheme, which is capable of data rates of 250 kbps, as well as a special operating mode capable of data rates of 625 kbps known as *turbo mode*. For further information about the uMRF board and its peripherals, please refer to Appendix C, Section C.2.

As depicted in Figure 5.2, the expansion port allows extending the functionality of the master station using a mezzanine board. Currently, three mezzanine boards have been devised: Wi-Fi, Ethernet and PNS. Prototypes of the Wi-Fi and Ethernet boards are shown in Figure 5.4. The main objective of these boards is to enable the master station with an alternative communication technology that can operate as a second tier for bridging the communication between the low-power network and the Web, for example. The Wi-Fi board is based on the ZeroG ZG2100M module, which is a single-chip IEEE



(a) Low-Power (b) High-Power

Figure 5.3: Master prototypes

Table 5.1: MRF24J40MA, MRF24J40MB and MRF24J40MC power consumption

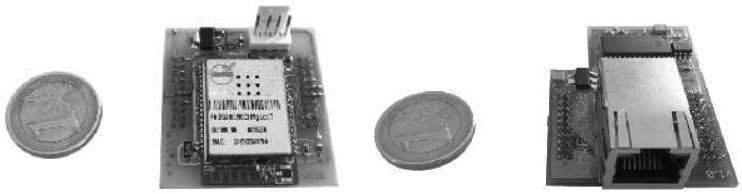
Mode	MRF24J40MA	MRF24J40MB	MRF24J40MC
Transmit	23 mA	130 mA	120 mA
Receive	19 mA	25 mA	25 mA
Sleep	2 μA	5 μA	12 μA

802.11b including MAC, baseband, RF and power amplifier, featuring hardware support for the AES and RC4 based ciphers (WEP, WPA, WPA2 security), SPI slave interface with interrupt and a serial trace interface (UART). The Ethernet board is built around the Microchip's ENC28J60 stand-alone controller, which is an IEEE 802.3 compatible Ethernet controller integrating a MAC and a 10Base-T PHY. Figure 5.2 also specifies CAN as a potential technology to include in the board. This could be important, for instance, to extend a FTT-CAN network with wireless communications.

As aforestated, the PNS aims at synthesizing interference and it is a key sub-block of the master station. The implementation adopted for the WFTT is the one based on the COTS components described in Section 3.4. Therefore, the PNS consists of a mezzanine board mounting several elements, the more important being the 32-bit MCU and the three MRF24J40MC transceivers. These encompass a Power Amplifier/Low Noise Amplifier and a Ultra Miniature Coaxial (U.FL) connector allowing to mount an external dipole antenna. Despite sharing the MAC/Baseband features of its MRF24J40MA and MRF24J40MB counterparts, the MRF24J40MC transceiver is characterized by a typical output power of +19 dBm with a 45 dB transmit power control range.

Software

The software architecture of the master station was designed on top of its hardware, i.e, the MCU for processing; the IEEE 802.15.4 transceiver for supporting wireless com-



(a) Wi-Fi (b) Ethernet

Figure 5.4: WFTT master mezzanine boards

munications; the PNS for synthesizing interference according to the WFTT protocol; the WITAS Event Logger for allowing the protocol's timeliness evaluation; and, finally, the human interface (HMI) for following the master's operation. Hence, on the bottom of the layered software architecture lies the hardware, as portrayed in Figure 5.5. The second layer corresponds to the hardware interface, including libraries to drive directly the MCU's UART and SPI ports, timers, and digital inputs and outputs. Additional libraries for controlling the PNS and the IEEE 802.15.4 MRF24J40MA/MRF24J40MB transceivers are also employed in this layer. These libraries are built upon the features provided by the hardware driver libraries mentioned before. Above the hardware interface is the WFTT layer, which makes use of the functions provided by the WFTT library to instantiate the WFTT protocol and facilitate the application development on top of it. The WFTT layer implementation is built around a timer with a 200-nanosecond resolution, enabling scheduling WFTT events with high temporal accuracy. This timer is used to control the lengths of the protected, contention and inactive windows, as well as the durations of the real-time slots in the protected window. Besides timer events, the WFTT layer implementation is driven by two other events: transmission and reception of a packet. In this sense, if any of these events occur, the flow of execution is passed to this layer, which proceeds to its analysis, checking if an action is required in response. This layer implementation relies heavily on the execution of ISRs to acknowledge, in a timely manner, timer expirations, packet receptions and packet transmissions. The current version of the WFTT layer does not provide a service to support the dynamic offset calculation of the real-time slave transmissions. These are statically defined at the application layer, where the requirements are set.

The WFTT layer includes an optional times-tamping library that was devised during the WFTT's early stages of development with the purpose of closely monitoring the timeliness of the events occurring during the execution of the WFTT protocol. Hence, when

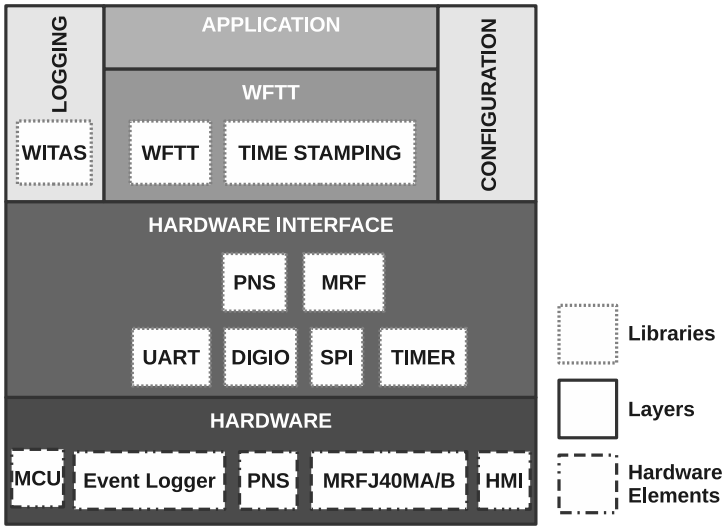


Figure 5.5: Software architecture of the master station

employed, it configures a 32-bit hardware timer to operate as a counter with a time resolution of 2 microseconds. It operates by time-stamping every event of the WFTT master state machine (beginning of the protected window; beginning of a real-time slot; and beginning of the contention and inactive windows) and, during the following WFTT inactive window, print the timing log to a terminal application running in PC via a serial port. The use of this library allowed to confirm the duration of the real-time slots and of the timing windows as they are perceived by the master station. Hence, it provided the necessary means to pre-evaluate the WFTT implementation, confirming the timely execution of the WFTT protocol in the master station.

The user application resides at the application layer. For the purpose of this work, the user application is very simple, since it mainly aims at providing a proof of concept of the WFTT protocol. As it will be detailed in Chapter 6, the user application considers a WFTT network with three real-time stations, each one performing a single transmission in every elementary cycle.

The logging cross layer represented in Figure 5.5 provides support for monitoring the performance of the WFTT protocol. This task is conducted using an external tool named WITAS, which collects information about the transmission and reception events occurring in a WFTT network. Using this information the WITAS tool is capable of computing the delays elapsing between the instants of packet transmissions and receptions. In order to cope with the WITAS operation requirements, it is necessary to embed the WFTT stations (master included) with an automatic mechanism that signals packet events and provides

access to its content. This is achieved at the logging cross layer. As documented in Figure 5.5, it employs the WITAS library, which encompasses the definitions and functions required to help the WFTT station interface the external WITAS tool. Additional information regarding the use of the WITAS tool to assess the WFTT protocol performance can be found in Chapter 6. A description of the WITAS tool architecture and operation is available in Appendix A.

The configuration cross layer was developed so as to render possible both changing the master's parameters in runtime, as well as controlling its operation. The parameters whose modification is supported, besides the *contention window* and *inactive window* durations, are the *sequence number* transmitted in the packets and used in the WITAS logging process; the communication *channel*; and the transmission *power* of the IEEE 802.15.4 transceiver. The control commands supported by this module are the *start/stop/reset* of the master station and the *start/stop* of the *black-burst* or *protective* interference transmitted by the PNS. The later commands were devised targeting the stand alone evaluation of the PNS. The master station configuration process is conducted by means of serial messages complying with a specific pre-defined format. These messages can be sent, for example, from a terminal emulator such as the HyperTerminal to the serial port connected to the master station. This software architecture description provides the necessary background for the study of the master's operation, which is presented in the following subsection.

5.2.2 Operation

Before detailing the operation of the firmware running in the master's station microcontroller, it is important to review the operation of the PNS, which supports the synthesis of both protective and *black-burst* interference. The PNS considered in the implementation of the master was detailed in Section 3.4 and operates in one of six states, namely:

Init: The three MCU SPI ports are configured to interact with the attached IEEE 802.15.4 transceivers in this state. Furthermore, the MRF24J40MC transceivers are configured to operate in the *turbo mode* and disable the CSMA/CA mechanism.

Hold: All the transceivers are idle, i.e., are ready to start transmissions but wait for the corresponding trigger signal. Also, their power amplifiers are turned off.

Armed: Paired transceivers perform transmissions in turns, on a selected channel and with a fixed period, which ensures that a given IEEE 802.15.4 channels can be

fully occupied. However, such transmissions are not propagated to the medium since all power amplifiers are turned off.

Black-burst noise: Packets are transmitted on the configured *black-burst* channel by two transceivers, in turns, provided that the associated power amplifiers are turned on.

Protective noise: Packets are transmitted in the configured protective channel by two transceivers, in turns, provided that the associated power amplifiers are turned on.

All: Packets are transmitted to the medium in both the configured *black-burst* and protective channels. This mode, however, is not available in the master implementation due to the hardware limitations identified in Section 3.4.

Although the PNS is not currently capable of simultaneously synthesizing both protective and *black-burst* interference, the master firmware implementation is not changed to cope with this limitation. This option is motivated by the fact that it will allow improving the PNS without requiring any modifications on the master's firmware. Furthermore, the PNS described in Section 3.4 is configurable and allows to define the synthesized interference type. Hence, for evaluation purposes, it will be possible to configure trials where the PNS transmits effectively one of the two types of interference. However, in the remaining of this section, it will be assumed that the PNS holds no limitations regarding the simultaneous support of both types of interference.

The operation of the master station can be modeled by a state machine encompassing eight states, as documented in Figure 5.6. When the master is turned on, it is set to the *INIT* state. The master remains in this state until all initializations are concluded, namely the initialization of the internal timer and inputs/outputs, of the PNS to the *HOLD* state and of the MRF24J40MA/B transceiver, among others. Regarding the transceiver initialization, there is a design option that is worth mentioning. Although the MRF24J40MA/B transceiver supports different security features, these were disabled in order to improve the time response determinism of transmitting and receiving packets. The future development of the WFTT protocol should account for the inclusion of security mechanisms to avoid unauthorized access to sensible information exchanged in the network.

Afterwards, the master switches to the inactive window (*IW*) state and programs the internal timer to expire at the end of its duration (IW_DUR). This allows the master to

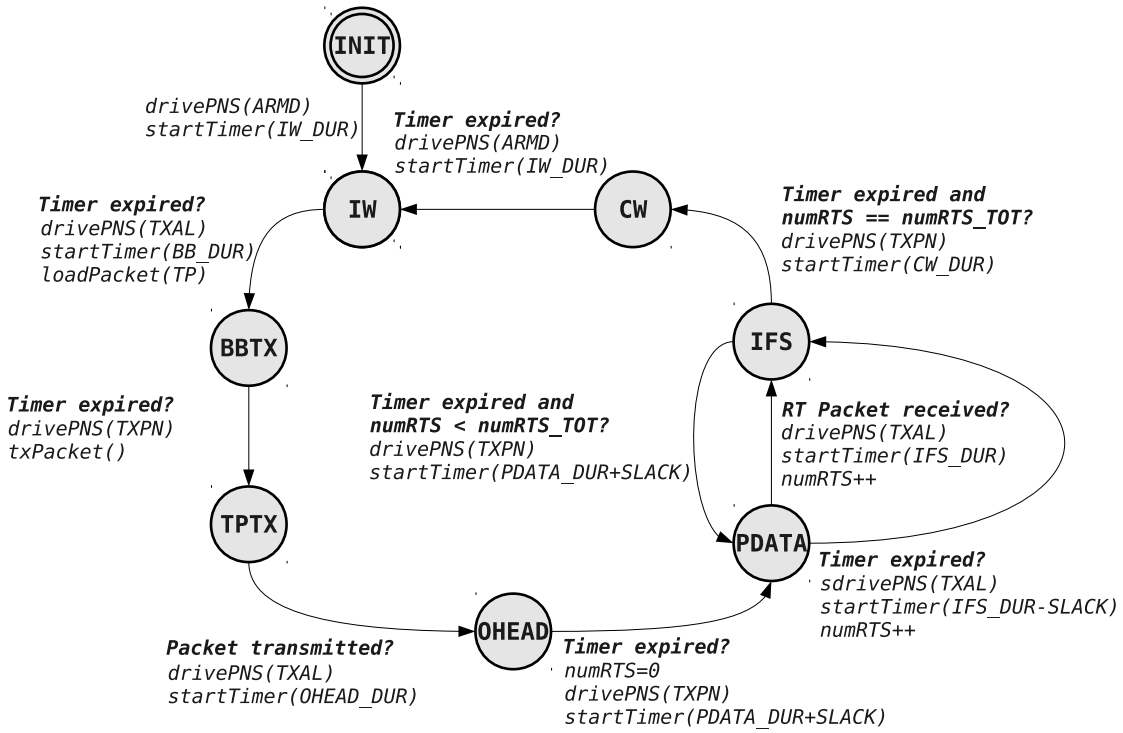


Figure 5.6: Master station's state diagram

enforce an inactive window with a (IW_DUR) duration. This state was devised to allow the medium access of "alien" technologies during the elementary cycle. In this state the master drives the PNS to the *ARMD* state where it becomes ready to transmit, i.e., although noise is locally synthesized by the PNS, it is not sent to the medium because the power amplifier is turned off.

When the timer expires, the master switches to the *black-burst* transmission state (*BBTX*), where it will eventually force the existing alien stations to backoff from the use of the medium. When this transition occurs, the timer is scheduled to expire after the duration of the ongoing *black-burst* (BB_DUR). The duration of this period is such that it guarantees an idle medium after its completion. In this state, the master drives the PNS to the *TXAL* state, ensuring that any "alien" stations contending for the medium will find it occupied in both *black-burst* and protective channels. Furthermore, to guarantee the shortest possible TP latency, the master loads the TP to the transceiver during this period, making it ready to transmit in the next state.

After elapsing the BB_DUR period, the master immediately initiates the transmission of the TP ruling the following elementary cycle. In addition, the master drives the PNS to the *TXKbytesPN* state for the duration of trigger packet. This interference pattern is

transmitted simultaneously to the TP, on a side channel, to ensure that alien stations will not find the medium idle and attempt transmissions, even with ongoing IEEE 802.15.4 transmissions employing a small amount of power. This is the *TPTX* state.

When the transmission ends, the master switches to the *OHEAD* state, which provides a time window where slaves can process the TP and become ready for transmitting without risking having the medium available for "alien" stations. This is implemented by occupying the medium with both *black-burst* and protective interference. During this period, the master sets the timer to expire after a period of time given by OHEAD_DUR and drives the PNS to the *TXAL* state. The OHEAD_DUR period corresponds to the sum of the slave's time to read the TP from its transceivers (5), process it (6), log it (7) and load a real-time packet to the transceiver for transmission (8), as presented in Figure 5.1.

After the overhead period elapsed, the master proceeds to ensure that the upcoming real-time transmissions are not affected by any "alien" stations. Hence, provided that their duration is known beforehand, the master switches to the *PDATA* state, configures its timer to expire after their duration (PDATA_DUR) plus a margin for slave jitter (SLACK) and drives the PNS to the *TXPN* state, ensuring that the ongoing real-time transmission by a slave will be secured against "alien" transmissions. In this state two scenarios can occur, both leading to the *IFS* state. In the first, the real-time transmission is received before the timer expires. In this case, the master increases the variable holding the number of real-time slots processed ($numRTS$), configures the timer to expire after the duration of the IFS (IFS_DUR) and drives the PNS to the *TXAL* state. In the second case, the timer expires before any real-time transmission has been received. This means that no real-time packet was detected during the protected window slot. Nevertheless, the master increases the number of real-time slots processed, drives the PNS to the *TXAL* state and configures the timer to expire earlier then the full length of the IFS_DUR, since it has waited an additional (SLACK) interval for the reception of the packet and it needs to be aligned with the elementary cycle phase.

The current version of the master's WFTT protocol implementation does not include a mechanism to enforce a medium busy state when a given scheduled real-time transmission is not issued as planned. The occurrence of an unoccupied real-time slot in the protected window creates a vulnerability, which can be exploited by "alien" stations initiating transmissions during this period, thus possibly compromising the remaining WFTT protected and contention window transmissions. However, given the early development of the protocol and the proof-of-concept nature of this work, this issue was not addressed in this

WFTT protocol implementation. The occurrence of an idle slot in the protected window can be be avoided by establishing a timeout for the corresponding slave transmission and, when not detected within this interval, by enforcing a busy state in the medium using the PNS to synthesize a *black-burst* interference sequence for the remaining of the real-time slot. The length of the timeout must be shorter than the minimum IFS or CCA durations of any contending technology.

The *IFS* state was devised to allow all stations participating in the WFTT protocol to be able to receive and process the real-time packets sent in the protected window. The duration of this state (IFS_DUR) was chosen to allow the slaves to read the received real-time packets from the transceiver (11) and log them accordingly (12), as documented in Figure 5.1. This state was also designed to hinder "alien" stations from capturing the medium during this interval by means of keeping the medium busy with protective and *black-burst* interference.

Following the *IFS* state, two paths of execution may succeed. In the first scenario, the total number of real-time slots ($numRTS\_TOT$) was not reached and, therefore, more real-time transmissions should follow. In this case, the master switches to the *PDATA* state. Hence, as before, it configures the timer to expire after the length of the real-time transmission (plus a margin to accommodate the slave jitter) and drives the PNS to the *TXPN* state. In the second case, the number of slots defined by the trigger packet was reached and, therefore, the master progresses to the *CW* state. Consequently, it configures the timer to expire after the length of the contention window (CW_DUR) and drives the PNS to the *TXPN* state.

After the timer expires at the *CW* state, the master returns to its first operational state, the inactive window (*IW*) state. As occurred when switching from the *INIT* state, the timer is configured to expire after its duration (IW_DUR). Furthermore, the master drives the PNS to the *ARMD* state again to ensure a high degree of interference readiness.

5.3 Slave Station

The slave station is the WFTT device that holds the application functionality, i.e., it is the element that actually communicates information pertaining to the application being developed. The WFTT master station is only responsible for coordinating the operation of the slave stations and, when required, bridging the WPAN with a second tier of communications. The slave, on the other hand, is bound to interface the application elements

(sensors, actuators, etc.) and to support the exchange of information in the network. The following subsections present the architecture and operation envisaged for the slave stations.

5.3.1 Architecture

Given the broad range of applications where a WFTT slave station may be employed, it is important to highlight its possible operation in different communication scenarios, ranging from applications with best-effort requirements to real-time applications. Hence, the slave's architecture and operation must enable a high degree of flexibility concerning its communication abilities.

Hardware

The hardware architecture is tightly coupled with the application being developed. Hence, it is common that the hardware choices made for the communication nodes in a network are dependent on the specific purpose they address. In the case of the WFTT protocol, the goal was the development of slave nodes that could be used for communication demonstration purposes and/or radio-frequency based localization. One of the key aspects driving the design of the slave station was the overall low power operation. It was originally intended that these nodes could operate with a battery for a period of, at least, two years in environmental monitoring applications. Hence, the choice of hardware accounted for this requirement. Figure 5.7 depicts the block diagram characterizing the slave station. As shown, a slave is built around a MCU, which is connected to the *Human Interface* (HMI), IEEE 802.15.4 *module*, *Power Control*, digital input/output pins (*DIO*) and analog input pins (*AI*). As an option, other elements can be integrated in the slave station board, namely: a *USB interface*, a *3-axis Accelerometer*, a *Temperature Sensor* and a Li-ion *Battery*.

The adopted MCU is the Microchip's PIC18F26K20 low power microcontroller. Although it employs the nanoWatt XLP Technology for reduced power consumption, it comprises a high-performance RISC CPU capable of 16 MIPS operation; a data EEPROM of 1024 bytes; a flash memory of 64 kB and 3936 bytes of RAM memory. Besides these key characteristics, the PIC18F26K20 can provide up to 35 input/output pins; two Capture/Compare/PWM (CCP) modules (one enhanced); one Master Synchronous Serial Port (MSSP) module; an Enhanced Universal Synchronous Asynchronous Receiver Transmitter

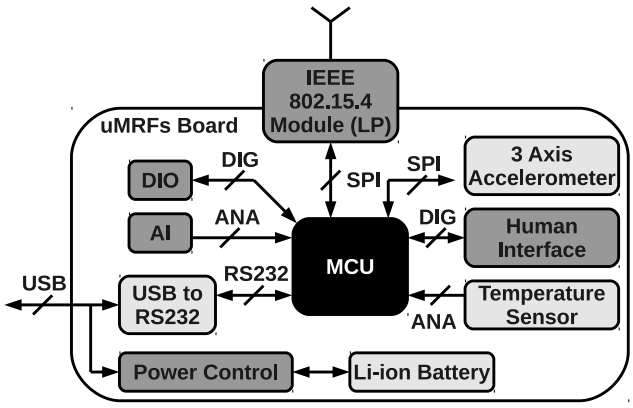


Figure 5.7: Block diagram of the slave prototype

(EUSART) module; and three 8-bit timers. The CCP module can be used in motor control applications, for example. The MSSP module has support for both 3-wire SPI and I^2C master and slave modes of operation. This module is configured as a SPI port to communicate with the IEEE 802.15.4 transceiver. The EUSART module is used to debug and configure the slave station, as will be presented further ahead. The timers can be used to count time or generate delays in an accurate fashion. This MCU allows the use of two of them together, forming a 16-bit timer. The firmware devised for the slave stations makes an extended use of this 16-bit timer to schedule events by enforcing pre-determined delays. Additional features of this microcontroller include an analog-to-digital converter (ADC) module with 10-bit resolution and 13 external channels, the support for brown-out reset with configurable voltages and a watchdog timer with programmable duration.

The HMI includes two LEDs (orange and green) and one push button. These elements are used to provide visual feedback about the slave's operation health and to control its run/stop state. The IEEE 802.15.4 module is the Microchip's MRF24J40MA device presented in Section 5.2.1. The ability to disable the CSMA/CA medium access mechanism is instrumental in the implementation of the WFTT protocol and one of the key features that motivated its adoption. The power control module allows the slave station to be powered by an external Li-ion battery, by the USB interface or both. In the last case, if the battery is not totally charged, the power control module charges the battery according to a predefined charging profile. The optional elements are only mounted on the board, when they are required by the specific application being developed. In this sense, the 3-axis accelerometer is used, for example, in applications requiring the monitoring of movements such as the survey of elderly activity patterns, for example. The temperature sensor may be used, for instance, in environmental monitoring. Figure 5.8 presents a photography of

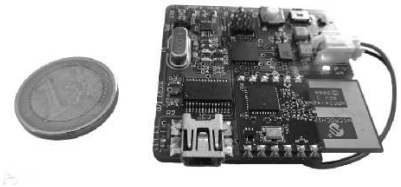


Figure 5.8: The slave prototype

the uMRFs board, which is used to implement the WFTT slave stations. For detailed information about the uMRFs board and its peripherals, please refer to Appendix C, Section C.1.

Software

The software architecture of the slave stations was designed following the same approach of the master station, except for the PNS, which is not required in slave stations. Hence, it is supported on a hardware layer encompassing the MCU, WITAS Event Logger, human interface (HMI) and MRF24J40MA transceiver external devices, as documented in Figure 5.9. The layer above (hardware interface) embodies the functionality of configuring and driving the slave's internal and external hardware. The hardware interface layer includes libraries to manage the operation of the MCU's UART and SPI ports, timers, and digital inputs and outputs. This layer also comprises a higher level library that builds on SPI communications to interface the slave's IEEE 802.15.4 MRF24J40MA transceiver.

Above the hardware interface lies the WFTT layer. As in the master station, this layer provides the necessary support for instantiating the WFTT protocol on the slave stations, making an extensive use of the WFTT library to facilitate the application development on top of it. However, this layer's implementation is significantly different from its counterpart in the master station, since the envisaged operation is also very distinct. In this sense, instead of providing support for scheduling real-time transmissions or protecting WFTT transmissions, for example, this layer implements mechanisms to control packet transmissions according to the WFTT protocol in both the protected and contention windows. Three types of events drive the WFTT layer: timer, packet reception and packet transmission. Timer events are programmed on a 16-bit MCU timer with a 250-nanosecond resolution and are used to accurately follow the elementary cycle timings defined by the trigger packet. An example of this timer usage is the scheduling of a real-time transmission in the protected window. Thus, upon the reception of a trigger packet, a slave station

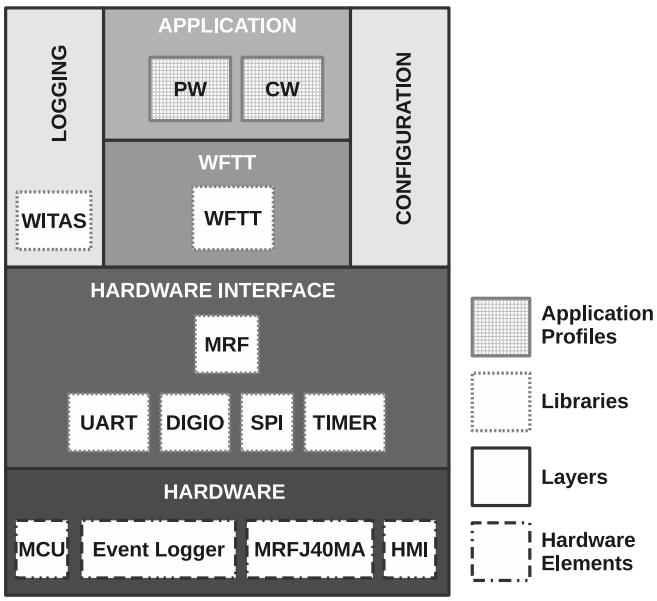


Figure 5.9: Software architecture of the slave station

scheduled to perform a transmission in the protected window loads its timer with the offset carried in the trigger packet. When the timer expires, the slave initiates its real-time transmission. Packet reception and transmission events caused by the MRF24J40 transceiver also affect the slave operation, as previously discussed. The WFTT layer relies on ISRs to manage the interrupts generated by the MCU timer and by the IEEE 802.15.4 transceiver.

As documented in Figure 5.9, the application layer is at the top of the slave's software architecture. The major difference regarding the master is the existence of application profiles, which allow a simple switching between slave modes of operation. The firmware of the slave station was designed to support an on-line modification of the type of traffic generated in each elementary cycle. Slave stations are usually programmed to exhibit an unchanged (real-time or contention) behavior during their lifetime operation. However, it was found to be of great great interest having an automated testing procedure that could change, in runtime, the type of traffic generated by the WFTT slaves. Following this requirement, two easily switchable application profiles were developed, each one generating traffic on one specific WFTT window: protected (PW) and contention (CW). Only one application profile can be active in any elementary cycle. If no application profile is enabled, then no traffic will be generated by the associated slave station. The switching between application profiles is supported by the configuration cross layer. The profile switching is always enforced in the subsequent elementary cycle following the switching request.

The WFTT PW profile should be enabled when the slave station is required to behave

as a real-time station, i.e., to perform transmissions in the protected window. Accordingly, the slave station will conduct a single transmission in its reserved slot as defined by the trigger packet. The timeliness assessment of slaves employing this type of traffic allows characterizing the WFTT protocol's support for real-time communications. If the WFTT CW profile is enabled in a slave station, it will become a standard-like IEEE 802.15.4 station, using the CSMA/CA mechanism to contend for the medium during the length of the WFTT contention window. The timeliness evaluation of the traffic generated in this profile renders possible the assessment of WFTT communications based on medium contention access with limited interference protection. Finally, the WFTT IW profile should be enabled whenever it is necessary to evaluate the performance of IEEE 802.15.4 transmissions without any protection from the WFTT protocol. This is useful to directly compare different WFTT types of traffic with regards to their timeliness and resilience in environments affected by "alien" interference.

Similarly to the master's software architecture (see Section 5.2.1), slave stations also encompasses both logging and configuration cross layers. The former enables the external WFTT protocol performance monitoring using the WITAS tool by providing information about the occurrence of packet transmission and reception events. The latter renders possible the runtime modification of parameters and control of the slave operation. The slave station supports the configuration of the IEEE 802.15.4 transceiver *channel* and transmission *power* parameters. The operation set up is managed using the *start*, *stop* and *reset* control commands. The configuration and control of a slave is conducted by means of serial messages complying to a specific pre-defined format. As before, these messages can be sent from any terminal emulator connected to the slave station's serial port.

Provided the background description of the slave's software architecture, the following subsection presents and analyzes its operation in detail.

5.3.2 Operation

As reported in the software architecture section above, the operation of a slave can be of two different natures, each one fulfilling a specific purpose. Real-time stations address the support of real-time traffic. Stations transmitting in the contention window enable best-effort traffic with some degree of protection against "alien" noise. Both types of stations have their transceiver's security features disabled in order to comply with the master's mode of operation. In the following paragraphs, each one of these slave operation types will be described in detail.

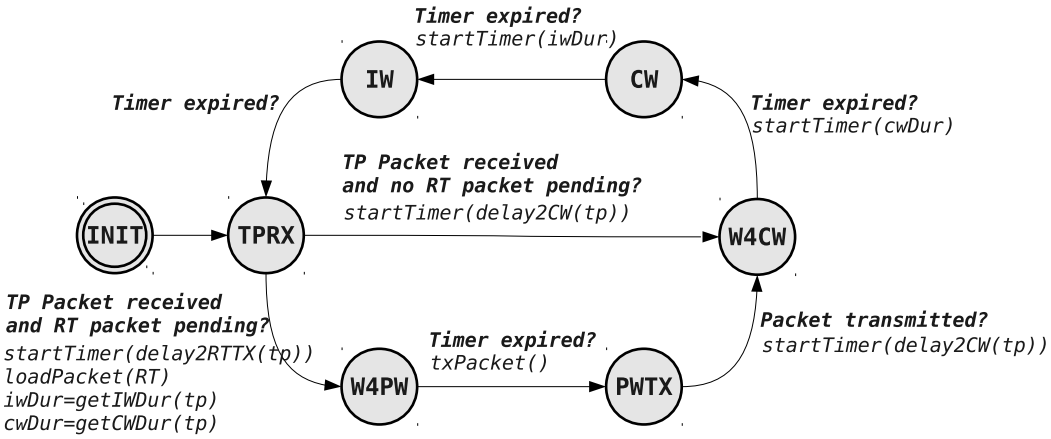


Figure 5.10: State diagram of the (real-time) slave stations transmitting on the PW

Real-Time Stations

As shown in Figure 5.10, the operation of slave stations as real-time nodes can be modeled by a state machine encompassing seven states. When the real-time slave is turned on, it is set to the *INIT* state, where it remains until the initialization process is over. During this period, among others, the internal timer and inputs/outputs, and the MRF24J40MA transceiver are configured. Then, the state machine progresses to the *TPRX* state in which it listens to the transmission of a trigger packet by the master station. When the TP is received, it is processed and one of two scenarios can occur. In the first scenario, the TP schedule does not refer the station. Therefore, the slave will configure the timer to expire in the beginning of the contention window and switch to the *W4CW* state. The delay programmed in the timer is obtained using the *delay2CW* function. In the second scenario, the TP schedule identifies the station and grants a real-time slot for its upcoming transmission. In this case, the station triggers the internal timer to expire at the scheduled offset, loads the real-time packet to the transceiver (to speed up its future transmission), obtains the length of the contention and inactive windows and proceeds to the *W4PW* state. The delay configured in the timer is obtained using the *delay2RTTX* function. In the *W4PW* state, the slave station waits for the real-time packet transmission instant to occur and, when it does, it triggers the transmission of the preloaded packet, proceeding to the *PWTX* state, which models the transmission of the real-time packet.

After transmitting the real-time packet, slave stations follow the elementary cycle timing without performing any additional transmissions. Hence, when the packet transmission is finished, the slave configures and triggers its timer to expire at the beginning of the con-

tention window (using the *delay2CW* function) and progresses to the *W4CW* state. When the timer expires at this state, the slave configures it again to delay for a *cwDur* interval, corresponding to the duration of the contention window, and progresses to the the *CW* state. Afterwards, the timer expires and the state machine progresses to the next state, the *IW* state, where it remains for the full length of its duration (*iwDur*). When the inactive window elapses, the state machine goes back to its initial operational state, the *TPRX* state, and repeats the steps described above.

Stations Transmitting in the Contention Window

Slave stations operating with contention transmissions can be modeled by a state machine encompassing six states, as documented in Figure 5.11. This state machine is similar to the one presented for the real-time stations in Figure 5.10, except for the packet transmission part. Hence, in the first operational state (*TPRX*), the station listens to the medium for a trigger packet. When this packet is received and decoded, the station configures its timer to expire at a random instant within the contention window. Therefore, the timer is programed with a delay value obtained from adding the contention window time offset obtained with the *delay2CW* function to a random delay value, generated with a uniform distribution ranging from 0 to the length of the contention window (obtained with the *getCWDur* function) minus the maximum duration of a contention packet transmission (*MCTX_DUR*). This duration is measured from the packet triggering instant to the instant where the associated transmission is concluded. Therefore, by using this upper bound for the transmission offset, it is possible to guarantee that the packet transmission will not violate its temporal limits.

After configuring the timer, enabling the CSMA/CA medium access mechanism with the *setCSMA* function (disabled by default in the WFTT protocol) and initiating the loading of the contention packet to the transceiver, the slave station progresses to the *W4CW* state. In this state, the station waits for the scheduled instant to perform the contention transmission. When this instant arrives, it triggers the transmission of the preloaded packet and progresses to the *CWTX* state, where it remains until the contention transmission is concluded. Afterwards, the slave station configures the timer (using the *delay2IW* function) to expire at the beginning of the inactive window and switches to the *W4IW* state. From here, as in the real-time station, the state machine goes to the *IW* state and, after, to the *TPRX* state, where its behavior repeats. When the station switches from the *W4IW* to the *IW* state, it configures the slave to operate back in the WFTT

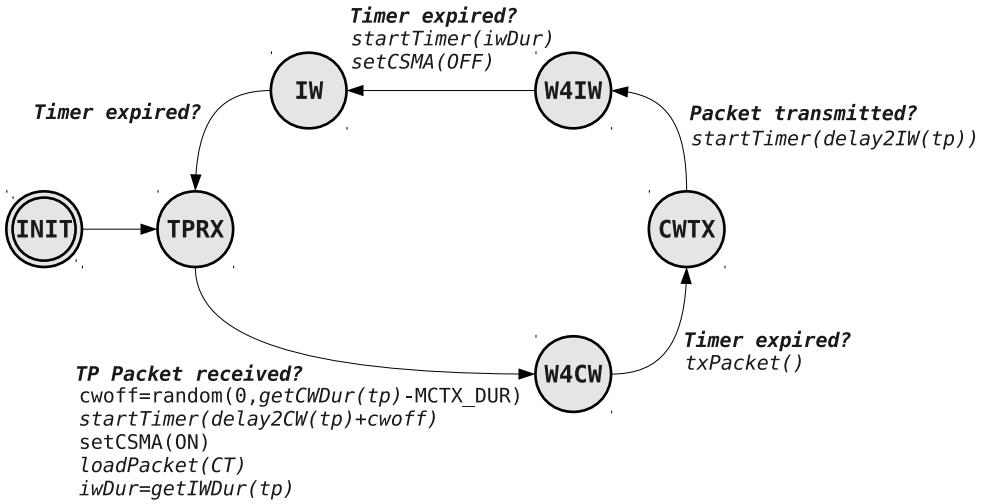


Figure 5.11: State diagram of the (contention) slave stations transmitting on the CW

default medium access mode (no contention) using the the *setCSMA* function.

5.4 Summary

This chapter presented the envisaged instantiation of the WFTT protocol. Hence, it explained the expected operation of the devices participating in a WFTT network, focusing on their interaction and timing behavior. In this regard, it was noticed that the IFS between consecutive data transmissions, required by the WFTT protocol, could be significantly reduced (or even eliminated) in specific scenarios. For example, when the packets transmitted by slave stations are not meant to be received by other neighbor slaves (only by the master) the IFS can be withdrawn from the WFTT operation. This occurs because the master can be continuously receiving these packets. In this case, the overall goodput of the network can be increased significantly.

This chapter also described both the architecture and inner operation of the two types of devised stations: master and slave. In this endeavor, the focus was on their hardware/software architecture and on the state machines that rule their operation. Regarding the state machine of the master, it was noticed that the WFTT protocol implementation described in this chapter does not yet support mechanisms to avoid idle periods in the protected window when a slave station fails to transmit its schedule packet. Given the proof-of-concept nature of the presented work, this issue was left to be addressed in a future version of the WFTT protocol implementation. Nevertheless, a simple mechanism

that seems adequate to mitigate this issue was described.

"The strongest arguments prove nothing so long as the conclusions are not verified by experience. Experimental science is the queen of sciences and the goal of all speculation."

Roger Bacon (1220 - 1292)

6 Protocol Assessment

In the previous chapter, the architecture and implementation of the WFTT prototype stations was presented and analyzed in detail. The following step is the timeliness evaluation of the WFTT protocol using the developed stations. Hence, in this chapter, the evaluation methodology is described with an emphasis on the devised testbeds and on the timing parameters of interest. The associated results are also presented and throughly discussed.

The approach employed to validate the WFTT protocol provides a solid evaluation of its timeliness and behavior. As it will be demonstrated, the evaluation was performed in two distinct scenarios. One in which the WFTT implementation was accomplished using conservative timing margins for coping with the jitter[1] of both the data transmissions and PNS synthesized interference, herein named *unoptimized WFTT implementation*, and the other where an effort was made to shorten the timing margins to the possible minimum, named *optimized WFTT implementation*. Both scenarios provide valuable information concerning the behavior and timeliness of the WFTT protocol.

In the first scenario, the trials were conducted on a residential building sub-basement (see Section 3.5) to characterize the WFTT protocol under varying conditions regarding the data and PNS transmission power. Besides identifying the protocol's timeliness issues regarding the timing margins, these trials allowed finding the best testing conditions, namely the distance between elements of the WFTT network and the optimal transmission power by those elements. As such, these results allowed focusing the analysis of the optimized WFTT version on the timeliness, emphasizing its noise immunity. In the second scenario the trials were designed to evaluate the WFTT timing behavior both regarding the

[1] In this dissertation the jitter is defined as the variable component of a delay, independently of the event nature (periodic or aperiodic) being measured.

master's trigger packet and the slaves' data transmissions. The slaves' related trials were focused on assessing the performance of their data transmissions on the WFTT protected and contention windows.

In order to assess the WFTT implementation, a measurement tool named WITAS was developed and employed in both unoptimized and optimized WFTT scenarios. The WITAS tool was designed to measure the key communication parameters of a wireless network comprehending several nodes. This tool supports the collection of parameters, namely the *number of successful transmissions*, *latency*, *energy* and can compute some related statistical variables such as the *minimum*, *maximum*, *average* and *standard deviation* delays, among others. Although this tool was developed to target wireless communication systems, it can also be used to assess wired setups. The WITAS tool supports time measurements with a 1 microsecond resolution and the assessment of small/moderate wireless networks.

A PC runs an application that holistically configures and controls the testbed, allowing a fast setup for each trial. This application can also be used to store and statistically process the (event) data collected at the Event Loggers. This information is sent by the Event Loggers to the Event Processor, which proceeds to its forwarding to the PC application. Detailed information regarding the WITAS architecture, operation and application can be consulted in Appendix A.

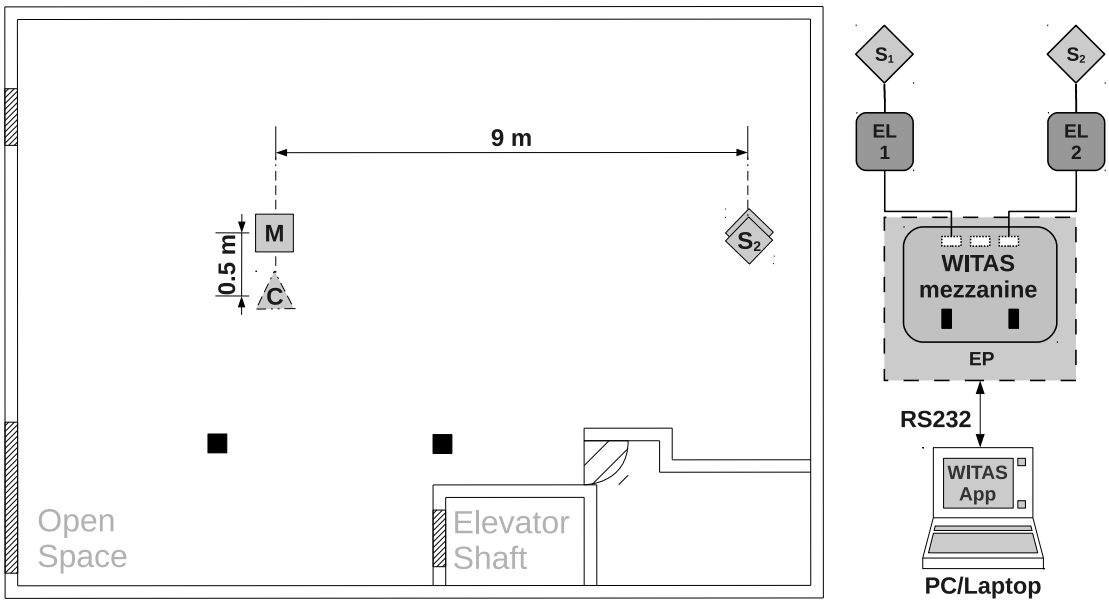
The following sections characterize the testbeds and the collected results for the unoptimized and optimized WFTT protocol implementations. In the latter case, the timeliness of the trigger packet transmissions and of the transmissions performed by the slave stations in the protected and contention windows are separately analyzed.

6.1 Unoptimized WFTT

The methodology employed in the evaluation of the WFTT protocol unoptimized implementation is presented in this section. Furthermore, the results obtained with the developed testbed are described and throughly discussed.

6.1.1 Methodology

Figure 6.1 depicts the main elements of the testbed used to assess the WFTT protocol timeliness in its unoptimized version. On the left (Figure 6.1(a)), the physical arrangement of the devices participating in the communications is illustrated. As documented, the test



(a) Physical arrangement (b) WITAS setup

Figure 6.1: Unoptimized WFTT setup

setup encompasses one master station (M); two slave stations (S_1 and S_2); and one Contender (C). The contender is placed in the vicinity of the master, separated by a distance of 0.5 meters. Studies [124] and [127] conclude that the placement of an Wi-Fi interferer near a IEEE 802.15.4 network generally yields a network with a significant PLR/PER. Hence, in order to evaluate such a challenging scenario, the contender's geographical location was selected so as to enable the assessment of the impact of having the interference source near to the key element of the WFTT network, the master. The slaves are placed at a distance of 9 meters away from the master, which is a value close to the standard nominal range of the IEEE 802.15.4 technology (10 meters). As documented in Table 6.1, the slaves are placed at different heights, namely 10 and 160 centimeters from the ground. This physical arrangement was adopted to determine if the slave height influences the WFTT performance.

Figure 6.1(b) depicts the connection of the WFTT elements to the WITAS evaluation tool. As it can be seen, the two slaves S_1 and S_2 are connected to the Event Loggers EL 1 and EL 2, which, in turn, are linked to the Event Processor (EP). The connection of the EP to the PC is made using a virtual serial port, which provides a bi-directional link between the control/process application running in the PC and the EP. The Event Loggers (ELs) timestamp the transmission and reception events occurring in the attached

wireless stations. After being timestamped, the events are collected by the EP in a round robin fashion. This procedure is conducted by means of command messages sent from the EP to the ELs. In response, data messages including event timestamp information are transmitted by the ELs to the EP. As introduced, the PC application obtains the timestamp information from the EP, allowing its storage and (later) analysis.

As detailed in Appendix A, a trial using the WITAS encompasses three phases: configuration, execution and data processing. The configuration phase represents the period of time in which all the elements participating in the trial are set up to operate during that trial. This includes not only the WFTT wireless stations, but also the elements of the WITAS. During this phase, each element is automatically assigned with new configuration that is sent by the WITAS application running on the PC. The configurations can be saved in text files, which can later be loaded and modified again, thus providing a high degree of flexibility. The approach of automating the trial setup allows saving a considerable amount of time in the configuration process.

The execution phase, where the trials are actually conducted, is started/stopped by the WITAS PC application. As before, this operation is conducted by sending serial commands to the EP for starting the round robin query process and to the master station to initiate the transmission of trigger packets, which begin the generation of communication events on the WFTT network. During the execution phase, the slave stations respond to the trigger packet and all events are logged by the attached EL devices and communicated upstream to the PC application. At the end of a trial, the collected data is temporarily stored on the WITAS application (memory), but can be permanently saved as a text file in a hard drive. This renders possible the loading of data to the WITAS application and performing its statistical analysis.

The data processing phase occurs when the user performs the statistical analysis of the event data currently on the applications' memory (be it after the end of a trial or after loading a text file with the trial's event data). In this phase, the collected events and their timestamps are used to calculate the communication's statistical parameters mentioned above. Besides these parameters, the WITAS PC application allows to automatically obtain the transmission latency histogram and to modify its bounds, thus changing its appearance. The results of the processing phase can be stored in text files for later reference.

The timeliness assessment of the unoptimized WFTT version was carried out using the general parameters shown in Table 6.1. Respecting the orientation defined in Figure 6.1(a), the contender was placed on a mobile MDF stand with a height of 64 centimeters,

Table 6.1: Unoptimized WFTT: general parameters

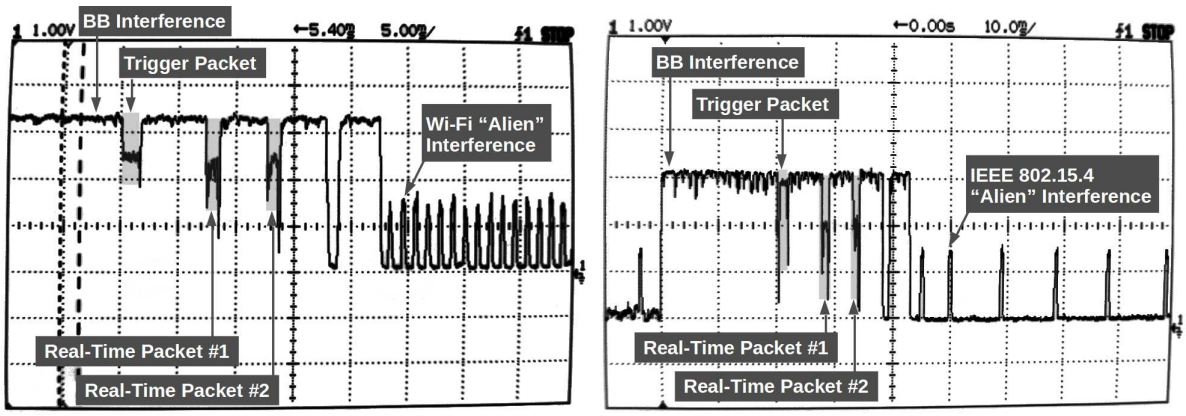
Parameter	Unit/Type	Contender		Master		Slaves
		CAOS	ZigFlooder	Data	PNS	
Height	cm	64		82		10 , 160
Channel	IEEE 802.15.4	—	14	14	11, 14	14
	IEEE 802.11	1	—	—	—	—
Power	dBm	20	0	0, -10, -20	-2, 18	0, -10, -20

the master on a MDF table with a height of 82 centimeters and the slaves of a custom made MDF stand, vertically aligned, at the heights of 10 and 160 centimeters, respectively. In order to assess the ability of the WFTT protocol to cope with Wi-Fi noise in an overlapping channel, both master (M) and slaves (S_1 and S_2) were configured to perform their data transmissions in channel 14, which shares a common region of the spectrum with the Wi-Fi channel 1. These transmissions were performed with three different power levels: 0 dBm, -10 dBm and -20 dBm. The motivation for supporting the use of multiple power levels is the evaluation of the WFTT protocol timeliness under different SIRs, allowing identifying possible bounds for degradation.

The master's PNS was configured to synthesize interference in either one of the 11 or 14 (IEEE 802.15.4) channels with a power of -2 dBm or $+18$ dBm. The signal synthesized by the PNS is transmitted on channel 11 if the master requires the broadcast of protective interference. Likewise, if *black-burst* interference is required, the signal is propagated on channel 14. As addressed before, both types of interference aim at populating the medium with energy that will, eventually, hinder co-located "alien" stations from either perceive the medium as idle or forcefully make them to backoff from the medium. As presented in Section 3.4, the implemented PNS can only synthesize one type of interference in a given instant. Hence, in the interest of evaluating the impact of both types of interference on the WFTT performance, individual trials were conducted in each scenario of PNS interference generation:

- Protective interference on IEEE 802.15.4 channel 11;
- *Black-burst* interference on IEEE 802.15.4 channel 14.

The contender station is used to simulate the medium contention that "alien" stations may impose on a WFTT network. Two interferers have been developed to meet this need:



(a) Wi-Fi (CAOS) (b) IEEE 802.15.4 (ZigFlooder)

Figure 6.2: Single contender "alien" interference examples

one based on the IEEE 802.11 technology, named CAOS, and the other, based on IEEE 802.15.4, named ZigFlooder. The former encompasses one USB Wi-Fi dongle configured to perform packet transmissions with a period of one millisecond on channel 1. This channel was selected because its overlaps the IEEE 802.15.4 channel (14) chosen to support the WFTT network. The Wi-Fi traffic is supported on packets with a length of 50 bytes (24 bytes for the PLCP and 26 bytes for the MPDU), which are transmitted at a rate of 1 and 2 Mbps (PLCP and MPDU, respectively) and at the dongle's maximum supported power (20 dBm). This traffic is capable of occupying a maximum of 29.6 % of the available medium time. The elected transmission period yields a small/moderate level of channel contention that enables the assessment of the WFTT effectiveness in avoiding Wi-Fi "alien" contention. Figure 6.2(a) illustrates the noise generated by the CAOS tool using a single USB Wi-Fi dongle. As it can be seen, the "alien" interference is only detected when both PNS interference and WFTT data packets are not being transmitted. For more information regarding the CAOS implementation, please refer to Appendix B.

The ZigFlooder is based on a uMRFs board programmed to transmit IEEE 802.15.4 standard packets on channel 14 with a length of 20 bytes, a power of 0 dBm and a period of ten milliseconds. These contention transmissions employ the "Energy Above Threshold" CCA mechanism with limit of -69 dBm. An example of this type of noise is depicted in Figure 6.2(b). The ZigFlooder channel was selected so that it overlaps the one used by the WFTT network. The chosen transmission period results in a maximum occupation of the channel time of 6.4 %, thus resulting in a low level of IEEE 802.15.4 "alien" contention to the WFTT network.

Table 6.2: Unoptimized WFTT: EC parameters

Parameter	Value
EC length	1100 ms
Slots in the Protected Window	3
Protected Window Length	40 ms
Contention Window Length	100 ms
Inactive Window Length	960 ms

Besides the trial's general parameters, the WFTT protocol elementary cycle parameters must also be configured before trials can be performed. The two slave stations are set up to operate as real-time stations conducting transmissions on their protected window designated slots. The master station was configured with the parameters documented in Table 6.2. As illustrated, the elementary cycle encompasses a protected window holding three reserved slots. The durations of the elementary cycle, protected, contention and inactive windows are 1100, 40, 100 and 960 milliseconds, respectively. The decision to use these values is motivated by the following facts:

- The elementary cycle duration is configured according to the requirement of the target application. Hence, it can be very short (e.g., 40 milliseconds) when the application merely requires a brief protected window for critical communications and the remaining (contention and inactive) windows can be suppressed, or it can be large in scenarios where the application demands a very low sampling frequency;

- The trials conducted for both WFTT implementations (unoptimized and optimized) require the use of a tool named WITAS, whose operation depends on the provision of event information by the WFTT network elements. This information is communicated by these elements to the Event Loggers during the inactive window. Hence, its length must provide enough time slack to accommodate these transactions. The value of 960 milliseconds was found suited for this purpose;

- Although the contention window is not used to perform any WFTT transmissions in the current scenario, a length of 100 milliseconds was reserved for completeness;

- Albeit only two slave stations are used in this unoptimized implementation testbed, the WFTT protocol was configured to reserve three real-time slots. Hence, the same

number of slots is used in both unoptimized and optimized WFTT implementations, which facilitates their comparison.

- As documented in Figure 5.1, the protected window encompasses several delays that are instrumental in guaranteeing the correct operation of the WFTT protocol. For example, besides the delay of loading the scheduled real-time (RT) packet into the transceiver, the overhead interval must account for the delay of reading, processing and logging the trigger packet (TP). In order to support all the required delays, the protected window was configured with a duration of 40 milliseconds. This value has been calculated by approximation, using estimates experimentally obtained.

The two slaves are configured as real-time stations 1 and 2. The use of only two stations is motivated both by a hardware limitation of the WITAS tool, which cannot support a high number of Event Loggers simultaneously, and by the fact that only two Event Loggers are required to ensure the proper timing consistency of the WITAS tool. The latter is a critical aspect that is assessed in every trial to guarantee that the timestamps used by the Event Loggers is properly aligned, i.e., that the Event Loggers have a common view of the time. This is validated by manually checking the timestamp values attributed by the Event Loggers to the "trigger packet receive" event, which occurs in every elementary cycle and must be simultaneously triggered and logged at both slave stations. If the timestamps of this event are synchronized among Event Loggers throughout a trial, then all the trial timestamps are valid. Otherwise, the trial is discarded. The latter case occurred only during the development and testing phase of the WITAS tool. However, for the sake of correctness, the "trigger packet receive" event timestamps were manually verified for all trials.

The use of two slave stations in a scenario where three real-time slots are reserved in the protected window has no foreseeable impact on the performed transmissions. This is supported by the fact that these stations occupy the first two slots of the protected window and, in theory, no "alien" station will be able to capture the medium before the end of their transmissions because the medium will not be found idle during that period. However, because the third slot is not used, an "alien" station may initiate a transmission that can extend to the contention window. In any case, since this testbed simply aims at assessing the timeliness of the two slaves operating as real-time stations, the occurrence of an "alien" transmission in the third slot or beyond is not relevant for the evaluation.

A trial is characterized by the occurrence of 1000 real-time transmissions. Provided that each EC comprehends two real-time transmissions, 500 ECs are required to complete

a trial. Hence, a trial is complete when 500 trigger packet transmissions (and the associated real-time packets) are performed. The evaluation of the WFTT real-time transmission timeliness is conducted for the three available PNS modes of operation: turned-off (OFF), transmitting *black-burst* interference (BB) or transmitting a protective interference (PI). The first case is used to collect reference information when there is no "alien" contender disputing the medium and no PNS interference to protect it. The second case is evaluated both under Wi-Fi and IEEE 802.15.4 "alien" noise. The third case is only assessed regarding Wi-Fi noise, as the PI generated by the PNS has no foreseeable impact on IEEE 802.15.4 "alien" transmissions. In all cases, the trials are conducted for three levels of transmission power (0 dBm, -10 dBm and -20 dBm), allowing evaluating the impact of this parameter on the WFTT performance. In order to assess the influence of the PNS power level on the WFTT ability to avoid "alien" noise, the trials conducted for the scenarios and cases mentioned above are assessed for two interference (BB/PI) power levels: -2 dBm and +18 dBm.

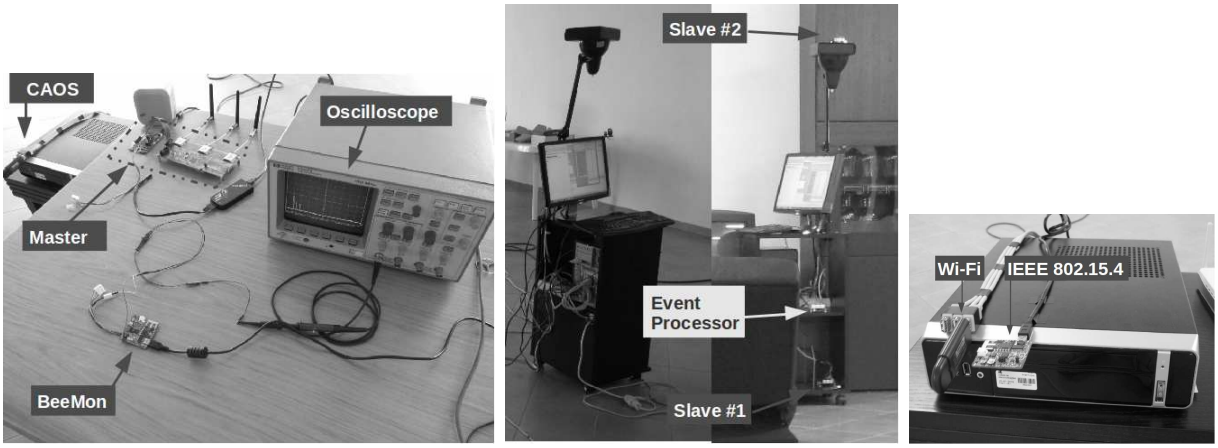
The raw results of each trial include the number of successful (SUCC) and failed (FAIL) real-time packet transmissions and the set of associated delays, besides the number of received trigger packets. Using the WITAS application, the minimum, maximum, average and standard deviation delay statistical parameters of a given trial are computed. Furthermore, employing an OpenOffice Calc spreadsheet, the real-time and trigger packet error rate are computed. The former corresponds to the percentage of real-time packets that were not successfully delivered while the later is the percentage of undelivered trigger packets.

Figure 6.3 depicts a set of photos of the testbed assembled to assess the performance of the WFTT unoptimized implementation. As documented, the photos show the master and slave stations that form the WFTT network, the BeeMon device that is used for monitoring the WFTT (data and interference) transmissions and, finally, the CAOS and ZigFlooder tools used to inject contention noise in the medium.

6.1.2 Results

The results of the WFTT unoptimized version assessment are presented in Tables 6.3 and 6.4. These tables document the WFTT timeliness results obtained when the PNS is configured to transmit interference with a power level of -2 dBm or +18 dBm, respectively.

The first aspect to consider in both tables is the interpretation of the error rate parameters. According to the proposed operation, for each trigger packet transmission there



(a) Master, BeeMon and CAOS (b) Slaves (c) CAOS and ZigFlooder

Figure 6.3: Unoptimized implementation testbed photos

will occur two logging events documenting its reception. Furthermore, when a given slave receives a trigger packet and realizes that it has a scheduled transmission on the protected window, it proceeds according to the described state machine, performing a real-time transmission in the designated slot. Hence, provided that the WFTT network encompasses two real-time slaves, there will be two real-time scheduled transmissions per trigger packet. The trigger packet error rate (TPER) is the percentage of trigger packets that were not received by any of the real-time stations. In other words, if a trigger packet is received by at least one station, it is assumed to be successful. The real-time packet error rate corresponds to the number of failed transmissions (FAIL) divided by the sum of the number of failed transmissions (FAIL) with the number of succeeded transmissions (SUCC).

Reference trials

In order to facilitate a comparative analysis of the results, a copy of the reference results described in Table 6.3 is also embodied in Table 6.4. As documented in these tables, the trials for WFTT data transmission power levels of 0 dBm and -10 dBm have very similar results. It is possible to observe that the real-time and the trigger packet error rates are null. Also, the transmission delay of the real-time packets is very close both in terms of its average value and its minimum and maximum values. The experimental real-time delay minimum and maximum values differ from their estimates, documented in Table 4.6, by approximately 3.2 % and 2.1 %, respectively, which is a small variation considering the assumptions used in their computation (e.g., assuming the *turbo mode* timings). Hence,

Table 6.3: Unoptimized WFTT timeliness with a PNS power of -2 dBm

PNS	Contender	Power (dBm)	SUCC (#)	FAIL (#)	RPER (%)	TPER (%)	MIN (μs)	MAX (μs)	AVG (μs)	STDV (μs)
OFF	None	0	1000	0	0.0	0.0	1530	1550	1541.9	4.75
		-10	1000	0	0.0	0.0	1530	1551	1542.1	4.68
		-20	967	22	2.2	0.0	1531	1550	1542.0	4.62
BB	Wi-Fi	0	122	25	17.0	84.8	1533	1550	1542.2	4.48
		-10	20	146	88.0	83.4	1535	1548	1543.0	3.94
		-20	6	122	95.3	87.0	1534	1548	1538.3	5.85
	802.15.4	0	860	41	4.6	6.0	1530	1550	1541.7	4.55
		-10	856	2	0.2	14.2	1531	1550	1541.7	4.63
		-20	900	4	0.4	9.4	1530	1551	1542.0	4.74
PI	Wi-Fi	0	998	1	0.1	0.0	1530	1551	1542.0	4.66
		-10	995	5	0.5	0.0	1530	1550	1541.7	4.76
		-20	579	208	26.4	0.8	1533	1550	1541.9	4.68

SUCC	*Succeeded Transmissions*	MAX	*Maximum Delay*
FAIL	*Failed Transmissions*	AVG	*Average Delay*
RPER	*Real-Time Packet Error Rate*	STDV	*Delay Standard Deviation*
TPER	*Trigger Packet Error Rate*	BB	*Black-burst Interference*
MIN	*Minimum Delay*	PI	*Protective Interference*

these results seem to corroborate the timeliness estimates' validity for the transmission of real-time packets.

Regarding the transmission power of -20 dBm, it is visible a small increase in the real-time packet error rate (RPER). Conversely, the trigger packet error rate (TPER) is kept constant at 0 %. This result suggests that the real-time packet errors are caused by a degradation of the SNR, motivated by both a reduction of the transmitted power and the relative position of the slave stations, which further contributes to decrease the received signal strength. Recall that the slave stations are vertically aligned at heights of 10 and 160 centimeters while the master is at, approximately, nine meters from the slaves at a height of 82 centimeters.

Trials secured by BB interference

In order to evaluate their immunity, the trials secured by BB interference are exposed to either Wi-Fi or 802.15.4 "alien" noise. The following subsections discuss the obtained results.

Table 6.4: Unoptimized WFTT timeliness with a PNS power of +18 dBm

PNS	Contender	Power (dBm)	SUCC (#)	FAIL (#)	RPER (%)	TPER (%)	MIN (μs)	MAX (μs)	AVG (μs)	STDV (μs)
OFF	None	0	1000	0	0.0	0.0	1530	1550	1541.9	4.75
		-10	1000	0	0.0	0.0	1530	1551	1542.1	4.68
		-20	967	22	2.2	0.0	1531	1550	1542.0	4.62
BB	Wi-Fi	0	176	25	12.4	77.8	1532	1550	1541.4	4.76
		-10	24	112	82.4	86.4	1532	1549	1539.0	4.15
		-20	15	115	88.5	86.8	1534	1549	1541.0	4.02
	802.15.4	0	874	63	6.7	0.0	1531	1550	1541.9	4.76
		-10	871	2	0.2	12.6	1531	1550	1542.2	4.77
		-20	854	2	0.2	14.4	1531	1551	1541.9	4.81
PI	Wi-Fi	0	997	1	0.1	0.2	1530	1550	1541.9	4.63
		-10	993	5	0.5	0.0	1530	1551	1542.0	4.65
		-20	770	112	12.7	0.6	1530	1551	1542.1	4.64

SUCC	*Succeeded Transmissions*	MAX	*Maximum Delay*
FAIL	*Failed Transmissions*	AVG	*Average Delay*
RPER	*Real-Time Packet Error Rate*	STDV	*Delay Standard Deviation*
TPER	*Trigger Packet Error Rate*	BB	*Black-burst Interference*
MIN	*Minimum Delay*	PI	*Protective Interference*

Wi-Fi "alien" noise

Consider the results depicted in Table 6.3 for the scenario where the PNS synthesizes *black-burst* interference with a power of -2 dBm. When a Wi-Fi "alien" interferer contends for the medium, the trigger packet error rate (TPER) significantly increases when compared to the reference values, reaching an average value of 85.0 % among the three power levels (0 dBm, -10 dBm and -20 dBm). This trend is closely followed by the real-time packet error rate (RPER) in the two lower transmission power levels (-10 dBm and -20 dBm). However, for 0 dBm, the RPER is much smaller than the one experienced by the TPER for the same transmission power. Also, it is important to notice that the -20 dBm case also yields a higher RPER, which is consistent with the reference scenario. Regarding the real-time transmission timeliness in this scenario, the minimum, maximum, average and standard deviation values have variations of less than 0.25 % when compared to their reference counterparts.

In the case of securing the WFTT network with +18 dBm interference, Table 6.4 reports a very high packet error rate for both trigger and real-time packets in the two lower levels of transmission power (-10 dBm and -20 dBm). However, following the trend observed in

Table 6.3, the RPER decreases to a moderate value of 12.4 % when the transmission power employed by the WFTT stations is 0 dBm. The TPER is also smaller in this case, but the error rate decrease is not so steep. The delay parameters of the real-time packets show a strong affinity with the results obtained in the reference trials. Indeed, the variation between the reference values and those obtained in the BB scenario for a Wi-Fi contender is bounded by a difference of approximately 0.2 %.

The trials presented in Tables 6.3 and 6.4 are characterized by a very high level of determinism concerning the delay timing parameters (minimum, maximum, average and standard deviation). This determinism is justified by the fact that the delay parameters are only calculated for successful transmissions and because the transmissions are performed employing a contention-free transmission mechanism. Thus, the transmission delay experienced by the real-time packets is mainly dependent of the hardware being used, which, in this case, does not result in significant changes on the delay statistical parameters.

The presented results also indicate that the *black-burst* interference produced by the PNS in both levels of transmission power (-2 dBm and +18 dBm) is not capable of avoiding high trigger and real-time packet error rates in environments affected by Wi-Fi "alien" noise. Assuming that the transmission power and distance between stations is adequate for supporting successful packet transmissions in the absence of interference (as noted on the reference trials), the high TPER/RPER can only stem from the corruption of WFTT data packets by external noise. This overlapping of data packets and "alien" noise can occur in two different ways:

1. The WFTT packet transmission is initiated concurrently to an ongoing "alien" transmission;

2. An "alien" transmission is initiated concurrently to an ongoing WFTT packet transmission.

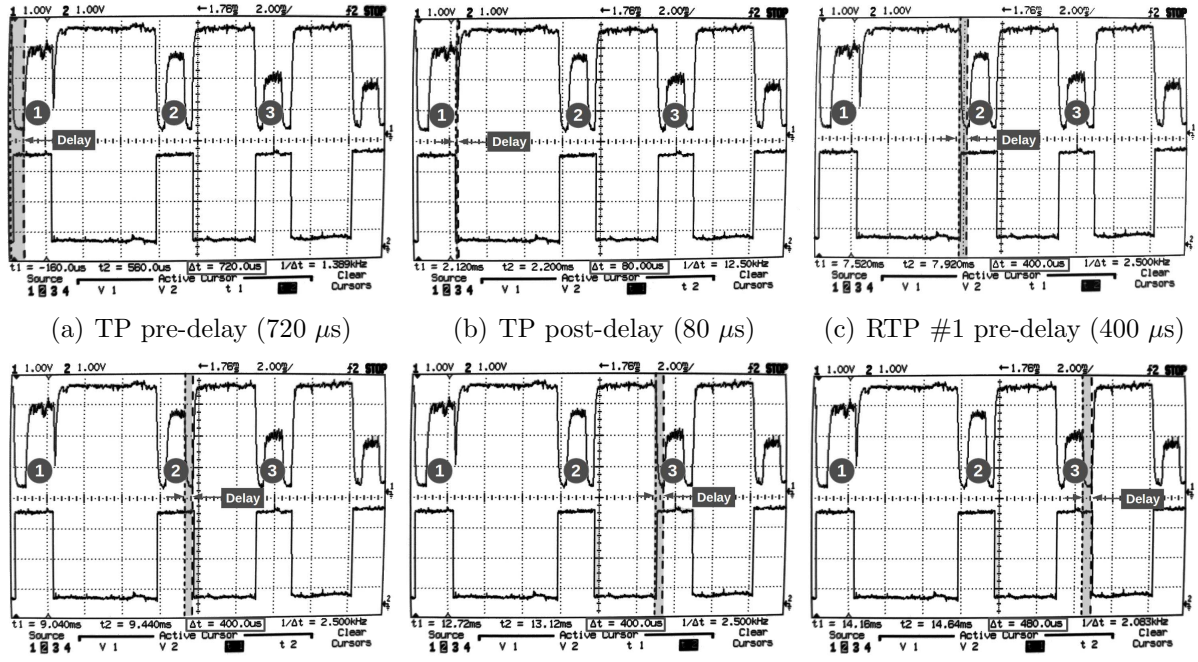
The first case occurs when the medium becomes free for an interval of time that allows the "alien" station to consider it idle and to initiate a new transmission. Since the WFTT protocol assumes that the real-time slots are free from "alien" noise, real-time stations do not check for a busy medium before initiating their packet transmissions. Hence, if an "alien" transmission is ongoing, a collision will follow. The second case occurs when the "alien" station perceives the medium as being free, even if a WFTT data transmission is being propagated. This issue may arise when the WFTT packet signal is not sufficiently strong to be acknowledged by the "alien" station.

The justification for a high TPER in this scenario seems to be mainly coupled to the first case. If the TPER was high as a consequence of Wi-Fi "alien" stations detected the medium idle when a data transmission was being performed, it would significantly increase for lower levels of data transmission power, which is not reflected in the results presented in Tables 6.3 and 6.4. Moreover, the trigger packets employing higher levels of transmission power are certainly detected by the "alien" stations, provided that the master station issuing them is geographically separated from the Wi-Fi contender by a distance of only 0.5 meters.

The RPER variation seems to be motivated by both cases of overlapping. Results in Tables 6.3 and 6.4 document a significant degradation of the RPER when the data transmission power falls from 0 dBm to -10 dBm or -20 dBm. Recall that the physical distance at which the slaves are from the contender is much higher than the equivalent distance to the master (9 meters *versus* 0.5 meters). These facts seem to support the conclusion that the amount of energy reaching the contender is not high enough for declaring the medium busy when the real-time packets are transmitted with power levels of -10 dBm and -20 dBm. Furthermore, if the real-time slaves perform transmissions at their maximum power (0 dBm), a significant RPER (e.g., 17 % in the PNS interference scenario with a power of -2 dBm) is still reported, possibly resulting from collisions with ongoing "alien" transmission on protected window reserved slots.

In order to better understand the results obtained for the unoptimized version of the WFTT protocol, a characterization of the transmissions during part of an EC was conducted using the BeeMon device detailed in Appendix D tied to an HP 54602B oscilloscope. Figure 6.4 shows the oscilloscope captures with overlay information identifying the measured intervals and specific parts of the signal marked with a number inside a circle (e.g., ②) for easier reference. These captures illustrate the trigger packet transmission ① and the first ② and second ③ real-time packet transmissions. The signal depicted between packet transmissions is the *black-burst* interference sequence. Although these captures where obtained using a WFTT network encompassing three slaves, the presented measurements only address the two first real-time transmissions.

Figure 6.4(a) demonstrates that the capture interval occurring between the BB sequence and the trigger packet transmission ① lasts 720 microseconds. Recalling that a medium idle period of more than 10 microseconds (see Table 4.4) may allow a Wi-Fi "alien" contender to declare it free and initiate a transmission, multiple "alien" transmissions can be triggered during the capture interval, which justifies the high TPER documented in both



(a) TP pre-delay (720 μs) (b) TP post-delay (80 μs) (c) RTP #1 pre-delay (400 μs)

(d) RTP #1 post-delay (400 μs) (e) RTP #2 pre-delay (400 μs) (f) RTP #2 post-delay (480 μs)

Figure 6.4: Unoptimized WFTT packet (measured) timings

Tables 6.3 and 6.4.

Results document a significant RPER of 17 % and 12.4 % for PNS transmission power levels of -2 dBm and +18 dBm, respectively. Although these values are notably smaller than those of the associated TPER, they still need to be addressed. The origin of these real-time packet errors can be traced to the impairment of the WFTT protocol timeline. This impairment occurs as the result of an "alien" transmission that is performed on a reserved interval, in one or more slots of the protected window, for example. Recalling that a contending "alien" station may initiate a transmission if the medium is sensed free for a given time interval, the packet errors are a result of the existence of relatively long idle periods between real-time transmissions in the protected window. This is illustrated in Figures 6.4(b), 6.4(c), 6.4(d), 6.4(e) and 6.4(f), which report the effective duration of time intervals between real-time transmissions. As depicted, the idle intervals range from 80 to 480 microseconds and were introduced in the (unoptimized) WFTT protocol implementation to provide a time margin for the PNS and MRF24J40 transceiver limitations identified in Section 3.4.2. As in the TPER case, these intervals are much longer than the identified 10 microseconds bound (Wi-Fi technology), which justifies the occurrence of "alien" transmissions in this period and the subsequent packet errors. However, one

observation arising from these results is that the time margin shortening is followed by a reduction on the packet error rate. In this sense, the optimization of the WFTT protocol implementation must account for a significant reduction of the time margins in order to reduce the probability of Wi-Fi "alien" stations being able to find the medium idle during that period.

Tables 6.3 and 6.4 allow a comparative analysis of the WFTT protocol timeliness for two levels of PNS interference power. Although results show a general trend of TPER and RPER reduction as a consequence of increasing the interference power from -2 dBm to +18 dBm, the variation does not seem very significant. An exception to this trend occurs for the value of the TPER when the employed data transmission power is -10 dBm.

IEEE 802.15.4 "alien" immunity

Consider the scenario where the WFTT network is protected by *black-burst* interference propagated with a power level of -2 dBm and is exposed to IEEE 802.15.4 noise. Compared to the reference trials, a moderate increase in the TPER is visible for all levels of transmission power. A TPER of 14.2 % is documented for the -10 dBm transmission power, followed by a 9.4 % for -20 dBm and 6.0 % for 0 dBm. Regarding the RPER, the highest value (4.6 %) occurs for a data transmission power of 0 dBm and decreases to almost zero for -10 and -20 dBm. The real-time transmission delay (minimum, maximum, average and standard deviation) suffers a maximum variation of 0.026 % when compared to the reference values. As documented in Table 6.3, the PNS BB interference is more effective in dealing with IEEE 802.15.4 noise than with Wi-Fi noise. This is mainly a consequence of the IEEE 802.15.4 longer idle interval required to declare the medium free (128 microseconds according to Table 4.5) and of the technology's CSMA/CA mechanism, which is more sensible to lower levels of energy.

Assume the scenario where the PNS synthesizes interference with a power level of +18 dBm. Compared to the reference trials, the value of the RPER experiences a small increase when the WFTT network is exposed to IEEE 802.15.4 noise. The exception occurs for the -20 dBm data transmission case, where it is actually reduced from 2.2 % (reference) to 0.2 %. However, for a data transmission power of 0 dBm, the RPER can reach the moderate value of 6.7 %. Regarding the TPER, it is null when the WFTT network employs data transmissions with a 0 dBm power level. Nonetheless, it increases to an average of 13.5 % in the two lower levels of data power (-10 and -20 dBm). Compared to case where the PNS interference is generated with a -2 dBm power level, the results show a small and inconsistent variation of the TPER and RPER. As in the Wi-Fi contender case, the real-

time packet timeliness does not suffer a significant variation (< 0.07 %) when the WFTT network is exposed to the IEEE 802.15.4 "alien" noise.

Although the WFTT immunity to IEEE 802.15.4 noise is higher than to Wi-Fi noise, the PER of trigger packets and real-time packets is not negligible. Regarding the TPER, as presented above, the time slack existing between the end of the WFTT *black-burst* sequence and the beginning of trigger packet transmission allows IEEE 802.15.4 based "alien" stations to find the medium idle and initiate new transmissions, thus overlapping the trigger packet and causing its corruption. However, because the clear channel assessment procedure of the IEEE 802.15.4 technology lasts 128 microseconds (refer to Table 4.5 for more information), which is significantly higher than its Wi-Fi counterpart, the TPER is lower. This results of a reduced probability of "alien" stations being able to find the medium free. Likewise, the existing slack intervals between the real-time transmissions in the protected window are long enough (see Figure 6.4) to allow IEEE 802.15.4 based "alien" stations to initiate transmissions during this period. However, as discussed above, the longer CCA and increased sensibility to noise contribute to a significantly reduced RPER, when compared to the Wi-Fi "alien" interferer case.

Trials secured by PI interference

The protective interference (PI) results presented in Table 6.3 (PNS power of -2 dBm) address the performance of trigger and real-time packet transmissions under contention from Wi-Fi "alien" stations. These results document a null TPER for WFTT data transmissions performed with 0 dBm and -10 dBm and a small TPER of 0.8 % for transmissions with a -20 dBm power level. The RPER is small for 0 dBm and -10 dBm, but increases to a significant value of 26.4 % when WFTT packets are propagated with a power level of -20 dBm. When compared to the reference values, the results collected in this scenario evidence a maximum deviation of 0.13 % regarding the delay timeliness parameters. Hence, there is a global indication that, by securing the trigger packet and the protected window with protective interference, it is possible to significantly improve the WFTT network immunity to Wi-Fi "alien" noise.

A similar trend is reported in Table 6.4 for the scenario where the PNS synthesizes protective interference with a power level of +18 dBm. In this case, the major difference is an accentuated drop in the RPER (from 26.4 % to 12.7 %) when the WFTT network operates with a power of -20 dBm. This variation seems to be a result of the increased protection provided by the higher power interference synthesized by the PNS. Otherwise,

the results are very close to those presented in Table 6.3.

Overall, Tables 6.3 and 6.4 provide evidence supporting the conclusion that the WFTT real-time packet transmission immunity to contention noise varies with the level of power being used in its propagation. For lower levels (e.g., -20 dBm), the RPER significantly increases. Although this trend is followed by the TPER, it is more severe for the RPER. Assuming that the "alien" transmissions can be effectively hindered by the protective interference, the justification for this effect is a reduction of the SNR, as a consequence of the lower signal levels. The reason why the RPER reduction is much steeper than the TPER is tied to the (relative) physical positioning of the stations, i.e, the slaves are placed perpendicularly at heights of 10 and 160 centimeters, while the master is located at a distance of 9 meters and at a height of 82 centimeters. The physical placement of the slaves is coincident with the physical region where each one's transceiver antenna exhibits the lowest gain, as illustrated by the radiation pattern of the MRF24J40MA datasheet [167].

Conclusions

The assessment of the WFTT protocol unoptimized implementation has disclosed several conclusions that must be accounted in the implementation of an optimized version. The first conclusion is that the time slack introduced in this implementation to cope with the timing limitations of both PNS and MRF24J40MA transceiver devices creates an opportunity for "alien" stations to perform transmissions in intervals reserved by the WFTT protocol for critical communications (trigger and real-time packets). This conclusion was based on the results obtained for the scenario where the WFTT network is protected by *black-burst* interference and exposed to IEEE 802.15.4 "alien" noise. In this case, the time slack is reduced by approximately half, when compared to the time slack reserved for the trigger packet (see Figure 6.4) and the IEEE 802.15.4 "alien" technology requires a longer idle period before initiating a transmission. The combination of both facts is determinant in significantly reducing the probability of IEEE 802.15.4 "alien" stations being able to find the medium free on the protected window.

Another conclusion is that the protective interference is highly effective in blocking Wi-Fi "alien" stations from accessing the medium and being able to initiate transmissions that jeopardize the WFTT protocol timeliness. This conclusion arises from the evidence that the RPER and TPER are very low in the presence of Wi-Fi noise, as long as the WFTT network is secured with protective interference.

The levels of transmission power employed by the WFTT network elements are addressed in the third conclusion. Results indicate a timeliness improvement trend when the levels of power used for both interference and data transmissions are higher. In this sense, regarding the implementation of an optimized version of the WFTT protocol, the PNS and the WFTT data transceivers should be configured with +18 dBm and 0 dBm levels of transmission power, respectively.

Finally, the delay statistics are highly consistent across all trials and all noise/protection/power scenarios. In fact, results report variations of less than 0.2 % for such scenarios when compared to the reference values collected in a noise-free environment.

Globally, the aforementioned conclusions allow stating that, if the two different types of interference (BB and PI) can be synthesized simultaneously, an optimized implementation of the WFTT protocol can support dependable real-time communications in open environments, where technologies such as Wi-Fi and IEEE 802.15.4 may contend for the medium.

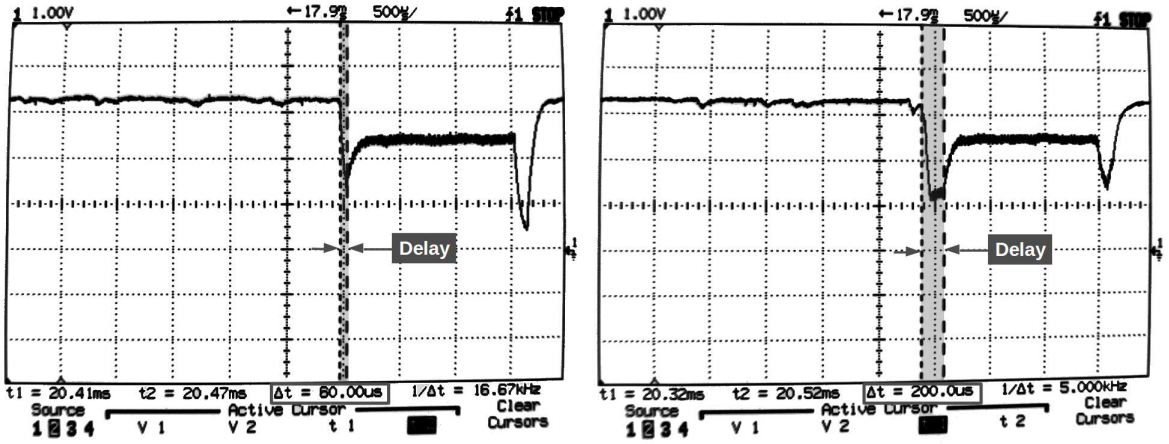
6.2 Optimized WFTT - Trigger Packet Timeliness

Following the results obtained for the unoptimized implementation of the WFTT protocol, this section analyzes a new version, herein named *optimized* WFTT implementation, which builds on the conclusions presented for the unoptimized version. This section is focused on the trigger packet timeliness and resilience to external interference, given its contribution to ensure the correct behavior of the WFTT protocol.

6.2.1 Methodology

The WFTT optimized version was created after realizing that the original implementation had issues related to the adopted safety timing margins. Hence, an effort was made to shorten these margins without compromising the overall WFTT behavior and timeliness. As it will be discussed, one of the key identified problems was the long timing margin reserved between the end of the *black-burst* transmission and the beginning of the trigger packet. This slack interval made it possible for "alien" stations to perceive the medium as idle, resulting in a high percentage of lost trigger packets and, consequently, in a significant number of elementary cycles without any transmissions.

The first step in optimizing the WFTT protocol was reducing the idle time between the *black-burst* and the trigger packet to a minimum, coping with both the jitter of the

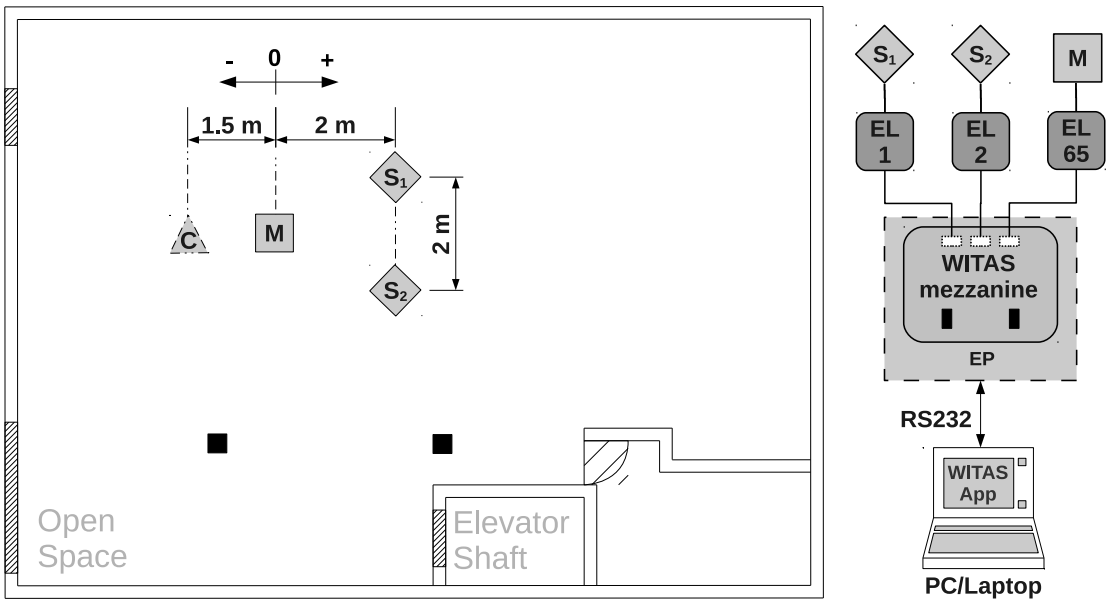


(a) Minimum delay (b) Maximum delay

Figure 6.5: Optimized WFTT capture interval length

Microchip's MRF24J40 transceiver and the jitter of the PNS. Hence, the optimized trigger packet timeliness was the first data flow to be assessed. Figure 6.5 illustrates two snapshots representing the minimum and maximum time margins of the implemented capture interval (between the *black-burst* sequence and the trigger packet). These captures were obtained using an HP 54602B oscilloscope fed with the output signal of a BeeMon device that was configured to operate on channel 14. As documented, the capture interval has a minimum of 80 microseconds and a maximum of 200 microseconds. Although these values are still significantly higher than the minimum SIFS (Wi-Fi) and CCA (IEEE 802.15.4), they are much smaller than those obtained in the unoptimized WFTT implementation (720 microseconds). Hence, a substantial improvement in the TPER is to be expected in both scenarios of Wi-Fi and IEEE 802.15.4 noise contention.

Figure 6.6(a) illustrates the physical arrangement of the setup used to perform this evaluation. As documented, the testbed encompasses one master (M) placed at the origin; two slaves (S_1 and S_2) placed in a perpendicular line at two meters from the master and two meters from each other; and the Contender (C) fixed at 1.5 meters away from the slave in the opposite direction of the slaves. The specific device arrangement shown in Figure 6.6(a) was motivated by several factors, namely the WITAS limitation of coping with large distances between Event Loggers and the Event Processor (see Appendix A), the need of evaluating the impact of an interference source close to all the elements of a WFTT network and the requirement of including the master station in the timeliness assessment. The latter is justified by the fact that, to compute the trigger packet delays,



(a) Physical arrangement (b) WITAS setup

Figure 6.6: Optimized WFTT: trigger packet timeliness setup

information about their reception and transmission instants is required. Since the master is responsible for performing the trigger packet transmissions, it needs to be connected to an Event Logger in order to register the corresponding instants.

Figure 6.6(b) depicts the connection of the WITAS evaluation tool to the elements that participate in the WFTT network. As illustrated, slaves S_1 and S_2 were connected to the Event Loggers EL 1 and EL 2, while the master (M) was tied to the Event Logger EL 65. All Event Loggers were configured with the same identification of their attached WFTT stations, as presented in Appendix A. Hence, the master was configured with the identification 65. The three Event Loggers were connected to the Event Processor, which was also tied to the PC running the WITAS application via a serial link. Although real-time packets were also transmitted/received by the WFTT stations in these trials, their information is filtered by the application so that the results only referred the trigger packet timeliness.

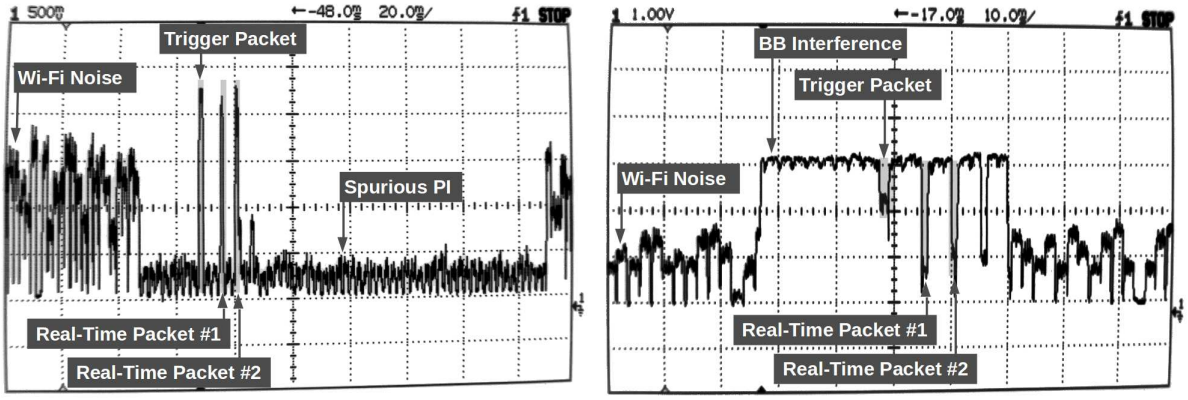
Table 6.5 depicts the general parameters of the setup used in this WFTT implementation assessment. As before, the contender was placed on a MDF stand with a height of 64 centimeters, the master on a MDF table at a height of 82 centimeters and the slaves over a large PVC table at a height of 74 centimeters, aligned as represented in Figure 6.6(a). The master and the slaves transmitted their data packets on channel 14 with a power of 0 dBm.

The PNS synthesized protective interference in channel 11 and *black-burst* interference in channel 14 employing, in both cases, transmissions with a power level of +18 dBm. The CAOS and ZigFlooder devices were used as "alien" medium contenders and were placed at a distance of 1.5 meters from the master. This position is herein designated as "@-1.5m". In the CAOS, five Wi-Fi USB dongles were set to perform packet transmission in channel (1) at the dongle's maximum supported power. The packets had an overall length of 50 bytes, 24 bytes for the PLCP and 26 bytes for the MPDU. The packets were transmitted at a rate of 1 (PLCP) and 2 Mbps (MPDU) with a period of one millisecond, resulting in a duty cycle of 29.6 %. The "alien" traffic produced by this CAOS setup provided a harsher contention environment when compared to the one used in the unoptimized WFTT implementation assessment, since five Wi-Fi dongles were setup in parallel to produce contention noise instead of only one. The option to increase the number of Wi-Fi stations emerged from the conclusion that the performance of an IEEE 802.15.4 network worsens with an increasing number of contending IEEE 802.11b stations [127].

Figure 6.7 shows two illustrative signals representing examples of the medium occupation of a WFTT network when the master's PNS synthesizes either protective or *black-burst* interference. These signals were obtained using the BeeMon device (see Appendix D) configured to operate on channel 14 and attached to an HP 54602B oscilloscope. The WFTT network used to obtain these signals was arbitrarily setup with one master and three slaves configured as real-time stations for demonstration purposes. Figure 6.7(a) depicts part of an EC where the WFTT transmissions are secured by protective interference. As documented, the Wi-Fi noise generated by the CAOS tool was only visible in the medium outside the bounds of the protective interference, which suggests that it is effective in hindering the Wi-Fi "alien" transmissions, even in environments with a very high level of contention. Figure 6.7(b) presents the part of an EC where the WFTT transmissions are guarded by *black-burst* interference. In this case, it is possible to observe that the Wi-Fi noise only occurred when either there were no ongoing WFTT data transmissions or *black-burst* interference, thus indicating that this type of interference is also suited to protect WFTT transmissions against Wi-Fi noise.

As in the unoptimized WFTT assessment, the interference produced by the ZigFlooder is based on the transmission of IEEE 802.15.4 standard packets on channel 14 with a length of 20 bytes, a power of 0 dBm and a period of ten milliseconds, resulting in a maximum occupation of the channel of 6.4 %.

The elementary cycle parameters used in this assessment are documented in Table 6.2,



(a) WFTT employing "protective" interference (b) WFTT employing "*black-burst*" interference

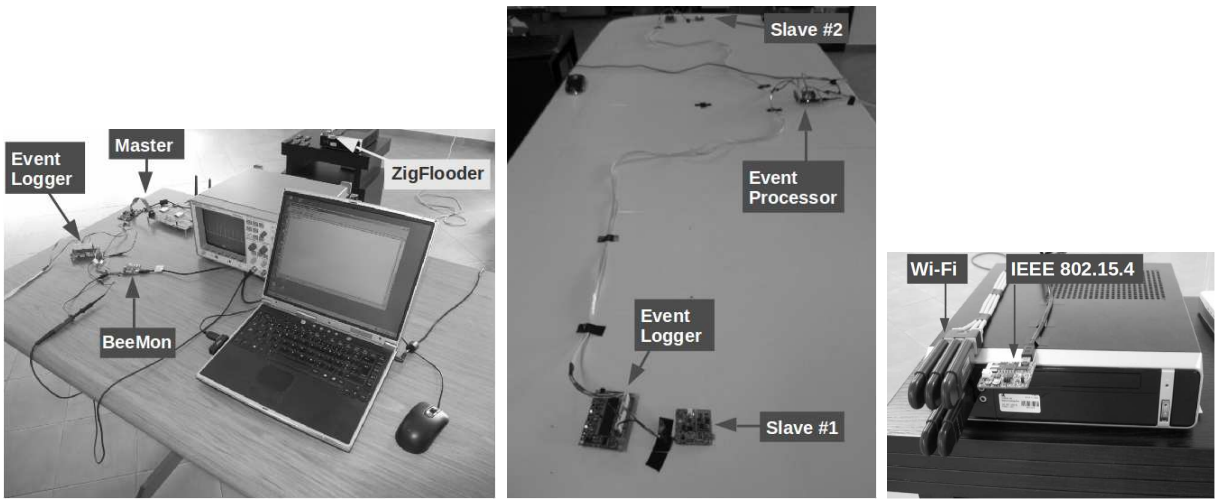
Figure 6.7: Wi-Fi multi-contender interference examples

Table 6.5: Optimized WFTT - trigger packet timeliness: general parameters

Parameter	Unit/Type	Contender		Master		Slaves
		CAOS	ZigFlooder	Data	PNS	
Height	cm	64		82		74
Channel	IEEE 802.15.4	—	14	14	11, 14	14
	IEEE 802.11	1	—	—	—	—
Power	dBm	20, 15, 16, 14, 20	0	0	18	0

i.e., the parameters employed in the unoptimized WFTT implementation were also used in this scenario. A trial was characterized by the occurrence of 1000 trigger packet transmissions. Provided that two slave stations were able to listen/log the trigger packet receptions, the execution of 500 ECs was required to to complete a trial, since each trigger packet originated two transmission logs. Similarly to the unoptimized scenario, the evaluation of the WFTT trigger packet transmission timeliness was performed in two PNS modes of operation: transmitting *black-burst* interference (BB) or transmitting a protective interference (PI). Both cases were evaluated in three perspectives: no "alien" noise, Wi-Fi "alien" noise and IEEE 802.15.4 "alien" noise. In all cases, the trials were conducted using a power level of 0 dBm in the WFTT data packet transmissions and +18 dBm for the PNS synthesized interference.

The raw results obtained in each trial encompass the number of successful (SUCC) and failed (FAIL) trigger packet transmissions/communications, besides the set of associated delays. A transmission delay is the time elapsing between the instant when the packet



(a) Master, BeeMon and CAOS (b) Slaves (c) CAOS and ZigFlooder

Figure 6.8: Optimized implementation: trigger packet testbed assessment photos

transmission is triggered and the instant when the packet reception event occurs at a slave. When a trigger packet is transmitted by the master, but it is not received by a given slave station, a transmission failure is said to have occurred.

Using the WITAS application, the minimum, maximum, average and standard deviation (delay) statistical parameters of a given trial were computed. Furthermore, using this application, it was possible to compute the number of (trigger packet) transmissions that were not received in all slave stations. This number was designated as "Zero Reception" failures because they correspond to the cases where the trigger packet is transmitted but not received by any of the slave stations participating in the WFTT network. To clarify this definition an example is provided. Consider a network composed by one master transmitting trigger packets and two slaves listening to them. Assume that two trigger packets are transmitted by the master in a trial. The first trigger packet is only received by one of the slaves, while the second trigger packet is not received by any slave. In this case, there is a total of three trigger packet transmission failures in which two of them are "Zero Receptions". The requirement of obtaining information about "Zero Reception" failures was motivated by the need of tracing their origin.

Figure 6.8 presents several photos of the experimental setup. As depicted, in addition to the stations composing the WFTT network (master and slaves), these photos illustrate the WITAS elements (Event Loggers and Processor) and the sources of "alien" noise that contend for the medium (CAOS and ZigFlooder).

Table 6.6: Trigger packet transmission timeliness (CAOS@-1.5m)

PNS	Contender Type	SUCC (#)	FAIL T(#) ≀ Z(#)	TPER (%)	MIN (μs)	MAX (μs)	AVG (μs)	STDV (μs)
BB	None	1000	0 ≀ 0	0.0	2109	2128	2116.0	2.74
	Wi-Fi	669	331 ≀ 6	33.1	2108	2127	2115.9	2.71
	802.15.4	1000	0 ≀ 0	0.0	2108	2127	2116.0	2.82
PI	None	1000	0 ≀ 0	0.0	2108	2128	2116.1	2.71
	Wi-Fi	970	30 ≀ 18	3.0	2110	2127	2115.9	2.92
	802.15.4	906	94 ≀ 94	9.4	2107	2127	2116.0	2.66

SUCC	*Succeeded Transmissions*	AVG	*Average Delay*
FAIL	*Failed Transmissions*	STDV	*Delay Standard Deviation*
TPER	*Trigger Packet Error Rate*	T ≀ Z	[T]otal ≀ [Z]ero Receptions
MIN	*Minimum Delay*	BB	*Black-burst Interference*
MAX	*Maximum Delay*	PI	*Protective Interference*

6.2.2 Results

Table 6.6 documents the trigger packet transmission timeliness results. In this evaluation setup, for each trigger packet transmission there will occur two logging events documenting its reception, provided that two slave stations participate in the WFTT network. Although these slaves will perform real-time packet transmissions according to the schedule defined by trigger packet, this information is filtered from the results, since this section focuses only on the trigger packet timeliness. The trigger packet error rate (TPER) presented in Table 6.6 corresponds to the percentage of unsuccessful (trigger) packet transmissions. If a reception event resulting from a trigger packet is generated, that transmission is said to be successful. The trigger packet error rate (TPER) corresponds to the total (T) number of failed transmissions (FAIL) divided by the total number of possible logged transmissions. In this case, since there are two slave stations, a total of 1000 transmissions should occur in an ideal scenario. As aforementioned, Table 6.6 depicts the total number of failed transmissions resulting from the simultaneous absence of trigger packet receptions in both slaves (Z).

Trials secured by BB interference

The trials secured using *black-burst* (BB) interference encompass the evaluation of the WFTT protocol timeliness in three different scenarios of "alien" noise contention: no noise, Wi-Fi noise and IEEE 802.15.4 noise. The following subsections discuss the obtained results

for each of these scenarios.

Noise-free scenario

As expected, there are no packet failures in the noise-free results documented in Table 6.6. Furthermore, when compared to the trigger packet delay estimated values presented in Table 4.6 (Section 4.4), the percentage variation is very small, of 0.86 % and 0.047 % for the minimum and maximum delays, respectively. These results confirm the validity of the corresponding estimates.

Wi-Fi "alien" immunity

The TPER experienced when the WFTT network is exposed to Wi-Fi noise is 33.1 % (see Table 6.6), which arises from a high number of lost trigger packets. However, the delay parameters show no significant variation as a consequence of the polluted environment where the WFTT transmissions are being carried out. In order to better understand the cause of this high TPER, the values of the total number of packets lost (T) and of the "Zero Reception" (Z) coupled lost packets are analyzed. As documented, for the possible 1000 successful transmissions, 331 where not properly performed and, from these, 6 were due to "zero reception" failures. This means that 325 transmissions failed because one of the two (packet reception) logging events associated to the transmission did not occur. Hence, for the majority of the transmissions, one of the slaves was able to receive the packet. This has not occurred only for three trigger packets, thus resulting in 6 packet failures reported.

Assume that the trigger packet transmission by the master experiences a similar attenuation in the path to both slaves, since they are geographically close and are placed at approximately the same distance from the master and from the CAOS contender. From the presented TPER results, it is possible to conclude that the high TPER is not mainly caused by the trigger packet corruption, since only three packets were effectively lost simultaneously by both slave stations. The high TPER suggests that the trigger packet failures are a result of "alien" transmissions within the capture interval. Such external transmissions are documented to have a much higher impact on one of the slaves, which indicates that the lack of packet errors in the other slave may result from a minimal noise power radiation in its direction. Because the WITAS tool does not offer information about the slave stations that logged the events, it is not possible to check which slave benefits from this CAOS transmission artifact.

IEEE 802.15.4 "alien" immunity

The trial where the WFTT network is exposed to IEEE 802.15.4 noise exhibits a null

TPER, which confirms that an "alien" contender is not able to access the medium when the capture interval is smaller than its required idle time for declaring the medium free. Furthermore, it also endorses the argument that the probability of finding the medium free when the capture interval is slightly higher than the "alien" minimum CSMA/CA is also small. In this case, the capture interval can reach the limit of 200 microseconds while the minimum CCA is only of 128 microseconds. Nevertheless, because an "alien" station response is highly dependent of the employed hardware, which defines its ability to transmit a packet immediately after the CCA period, the observed behavior may not be replicated by "alien" IEEE 802.15.4 stations employing a different transceiver.

Regarding the delay parameters, results also corroborate that the "alien" IEEE 802.15.4 interference has no meaningful impact on the trigger packet timeliness when the WFTT network is secured by *black-burst* interference.

Trials secured by PI interference

As in the previous case, trials secured by protective interference (PI) are conducted in three distinct scenarios of "alien" noise contention: no noise, Wi-Fi noise and IEEE 802.15.4 noise. The obtained results in each scenario are analyzed in the following subsections.

Noise-free scenario

On par to what occurred in the trials secured by BB interference, Table 6.6 demonstrates that no packet failures were detected in the noise-free trial. Compared to the trigger packet delay estimated values presented of Table 4.6, the variation in percentage is still very small, i.e, of 0.81 % and 0.047 % for the minimum and maximum delays, respectively.

Wi-Fi "alien" immunity

In the scenario where the WFTT network is secured by protective interference against Wi-Fi "alien" contention the TPER rises to 3.0 % when compared to the reference value. This value results from the overall occurrence of 30 trigger packet failures, in which 18 are caused by a failure to receive the trigger packet in both slave stations. This means that 1.8 % of the failures where due to the trigger packet not being received by any of the slaves and the other 1.2 % were originated by a reception failure in only one of them. This seems to be caused by the occurrence of "alien" transmissions within the capture interval. Recalling that the CAOS tool employs five USB Wi-Fi network adapters that attempt to perform transmissions at every millisecond and that each adapter has a specific radiation and sensitivity pattern, it is possible that the protective interference has different

hindering impacts on each CAOS Wi-Fi adapter. Hence, it is feasible that in most trigger packet transmissions, the master's PNS protective interference is highly effective in blocking the transmissions from a given CAOS Wi-Fi adapter while in a few there is an "alien" Wi-Fi adapter whose sensitivity to the protective interference is reduced and an "alien" packet transmission is initiated during the capture interval. This explains the occurrence of trigger packet reception failures at both slaves. A possible solution for this issue can be the use of new WFTT devices, which are able to transmit interference synchronized with the master station. If such devices were deployed around the physical area of the WFTT network, as suggested for mitigating the hidden node problem in Section 4.2.3, the synthesized interference visibility by "alien" stations would increase, thus reducing the issue's probability of occurrence.

Likewise, as observed above, the occurrence of reception failures on a single slave can arise when the radiation pattern of the Wi-Fi adapter has a minimum in the direction of a slave. In this case, although the "alien" transmission is performed, the level of energy reaching the slave is not high enough to lower its SNR to a level where the packet reception fails. Regarding the delay parameters, no significant variation is shown. Overall, considering the harsh contention environment created by the CAOS tool in this trial, the TPER values are very small.

IEEE 802.15.4 "alien" immunity

In this scenario, the protective interference has no meaningful impact on blocking the "alien" noise since the PNS interference is not sensed by the IEEE 802.15.4 "alien" station. Hence, the TPER rises to a value of 9.4 %, where all packet failures are caused by a corruption of the trigger packet due to the overlapping with an ongoing "alien" transmission. This observation is supported on the fact that all trigger packet failures occurred simultaneously in both receivers. Hence, the packet decoding failed during the reception process, which is a strong indication that it was corrupted. Despite the protective interference inability to avoid "alien" transmissions, the TPER is not very high when the WFTT network is exposed to "alien" IEEE 802.15.4 noise. This is the result of using a small duty cycle for the transmission IEEE 802.15.4 "alien" noise. If this duty cycle increases, the medium contention will be higher and the probability of collisions will also increase. As in the previous cases, the delay parameters show no significant variation when compared to the reference values (noise clear environment).

Conclusions

The analysis of the trigger packet timeliness has allowed an experimental verification of the improvements resulting from optimizing the WFTT protocol implementation. The major conclusion is that a combination of *black-burst* and protective interference is capable of ensuring a very high level of protection against both Wi-Fi and IEEE 802.15.4 "alien" noise. Results demonstrate that the TPER is very small (3.0 %) when protective interference is used to secure the WFTT network. Also, there is evidence that the use of *black-burst* interference is effective in avoiding contention noise from IEEE 802.15.4 "alien" sources. Although these results are not optimal in the sense that a 3.0 % TPER is not negligible and that the null TPER for the IEEE 802.15.4 "alien" noise was obtained in a contention environment with a low duty cycle, the time margins can be further reduced, thus potentially enhancing the overall WFTT network response to "alien" noise.

Another conclusion arises from the consistent values of the delay parameters obtained in all trials. As documented, the minimum, maximum, average and standard deviation delays show no significant variation throughout the trials, regardless of the existence of noise and the type of noise used. This is an expected result emerging directly from the adopted contention-free transmission scheme and from the medium protection against "alien" noise of the WFTT protocol. In summary, by performing transmissions and employing *black-burst* and protective interference according to the WFTT protocol definition, it is possible to ensure the transmission of trigger packets with a high level of determinism for both delay and error rate, even in challenging contention environments.

6.3 Optimized WFTT - Slave Timeliness

After experimentally validating the improvements on the trigger packet timeliness arising from optimizing the WFTT protocol implementation, it is necessary to assess the performance of the different types of traffic that can be supported on an elementary cycle. The methodology employed to evaluate the different types of traffic and the associated experimental results are presented and analyzed in the following subsections.

6.3.1 Methodology

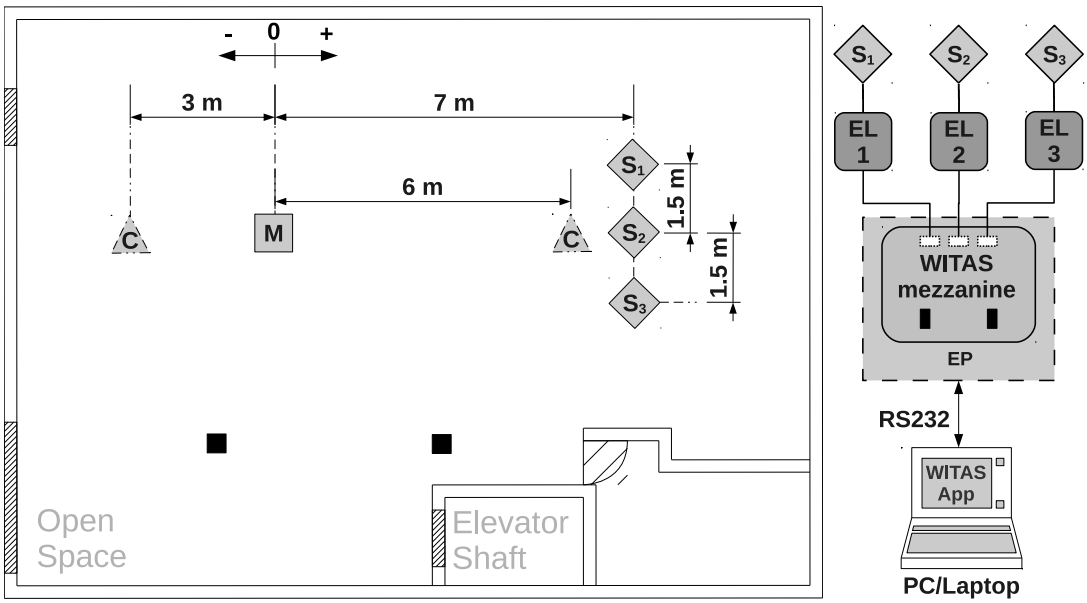
In the previous section, the WFTT protocol assessment setup was focused on the trigger packet timeliness. This section also addresses the optimized WFTT implementation, but

targets the slave station timeliness, i.e., the timeliness of the real-time packets, transmitted on the protected window, and of the contention packets, transmitted on the contention window. Slave stations were configured to perform only one type of transmission per elementary cycle. Real-time transmissions were performed on the reserved slots of the protected window while contention transmissions were conducted at one random instant within the contention window. Due to the PNS limitations described in Section 3.4.2, real-time transmissions were secured by either protective or *black-burst* interference in a given elementary cycle, as in the previous assessment. Transmissions in the contention window were secured only by protective interference.

The setup used to evaluate the timeliness of the slave transmissions is pictured in Figure 6.9(a). As documented, it is composed by one master (M) placed in a MDF stand at the origin; three slaves (S_1, S_2 and S_3) placed on a PVC table, in a perpendicular line to the master (1.5 meters apart of each other) and at a fixed distance of approximately 7 meters from it; and the Contender (C) placed on a MDF mobile stand that can be moved into two different locations: one at 3 meters form the master (herein designated as the position "@-3m") and the other at a distance of 6 meters from the master in the opposite direction (herein designated as the position "@+6m"). The spacial arrangement devised for this assessment was driven by the results obtained in the two previous evaluations, having the contender close to the master. In this case, we chose to put the contender at more reasonable distances both from the master and from the slaves to draw conclusions regarding its impact on the trigger packet error rate and on the error rate associated to packets transmitted by the slaves. The distance of 1.5 meters between slave stations was motivated both by the WITAS distance limitation between Event Loggers and the possibility of evaluating a deployment scenario where the WFTT network covers a small geographical area.

The use of a three slave station setup provides a more realistic network setup to compare the transmissions timeliness of the WFTT protocol in its protected and contention windows. As explained before, a packet transmission delay corresponds to the time elapsing between the instant of the packet transmission trigger command and the instant of the packet reception event at a given slave.

Figure 6.9(b) illustrates the WITAS connection to the WFTT elements used in the testbed. Since the assessment targets the evaluation of the slave station performance, only the slaves were connected to the Event Loggers and registered their events for the subsequent timing analysis. The WITAS application was used to configure the testbed



(a) Physical arrangement (b) WITAS setup

Figure 6.9: Optimized WFTT: slave packet timeliness setup

parameters, store the collected data and process it to obtain the statistical timeliness results of the transmissions performed by the WFTT slave stations. The parameters employed in this evaluation were common to the trigger packet timeliness evaluation scenario. Hence, they are also expressed by Table 6.5.

The EC parameters adopted for each one of the (protected and contention window) transmission scenarios is documented in Table 6.7. As depicted, the EC reserved three slots in the protected window in all scenarios. However, these slots were only effectively used in the scenario where slave stations were configured to perform real-time transmissions, i.e., in the "Protected Window Transmission Scenario". The length of the protected window was 40 milliseconds, providing enough time to cope with the delays that ensured a correct operation of the WFTT protocol, as represented in Figure 5.1. In this scenario, the events occurring in the protected window were uploaded to the attached Event Logger during the inactive window, which provided an interval of 860 milliseconds for this purpose. The length of the contention window was arbitrarily selected to be of 100 milliseconds and could be reduced if required.

The duration of the contention and inactive windows is dependent of the type of packets whose timeliness is being assessed. In the "Contention Window Transmission Scenario", slave stations were configured to exclusively perform transmissions on the contention win-

Table 6.7: Optimized WFTT: EC parameters

Parameter	PWTS	CWTS
EC length	1000 ms	1400 ms
Slots in the Protected Window	3	3
Protected Window Length	40 ms	40 ms
Contention Window Length	100 ms	800 ms
Inactive Window Length	860 ms	560 ms

PWTS - *Protected Window Transmission Scenario*
CWTS - *Contention Window Transmission Scenario*

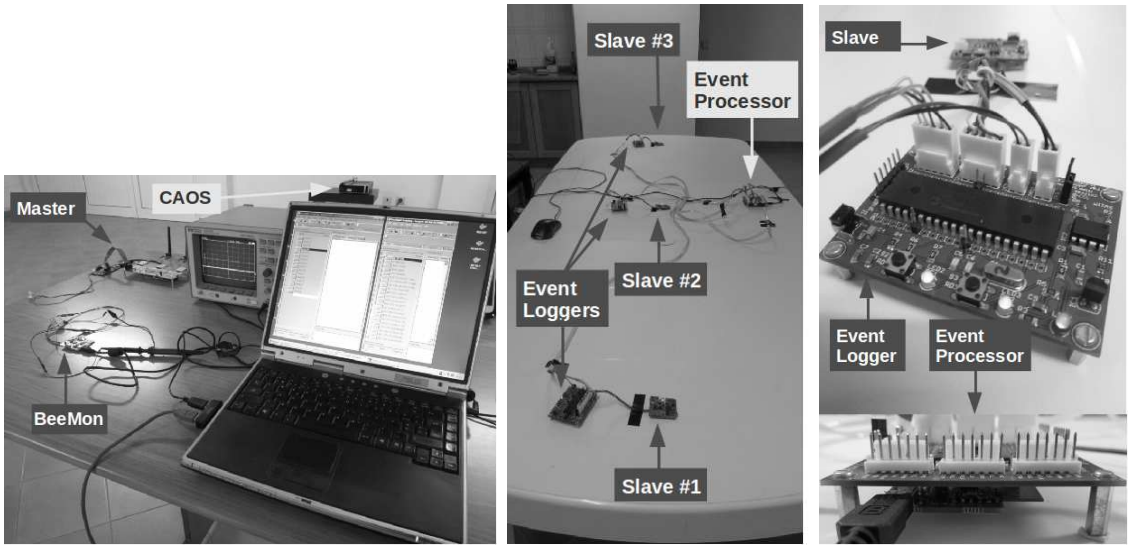
dow. In this case, the contention and inactive windows were configured to have a duration of 800 and 560 milliseconds, respectively. These intervals provided enough room to randomize the transmission triggering instant on the contention window and to upload all event information to the attached Event Logger during the inactive window.

A trial is characterized by the occurrence of 1000 packet transmissions. Provided that three slave stations were able to listen/log the corresponding packet receptions, the execution of 167 ECs is required per trial, as each trigger packet originates six transmission logs (one packet transmission originates two receptions, hence two transmission logs). As previously stated, the timeliness of the WFTT protocol was evaluated using two different types of packets:

Real-time: packets transmitted on the protected window in designated slots. This type of traffic was used to assess the WFTT protocol ability to support dependable real-time communications;

Contention: packets transmitted at a random instant within the contention window. The WFTT protocol ability to support best-effort communications was tested using this type of traffic;

The "Energy Above Threshold" CCA mechanism (CCA1 mode) was adopted with a -69 dBm medium busy threshold for all best-effort transmissions, since, according to Bertocco *et al.* [108], it is the worse selectable CCA mechanism to avoid Wi-Fi noise. As before, the assessment was conducted using either *black-burst* (BB) or protective interference (PI) with a power level of +18 dBm and WFTT data packets with a 0 dBm transmission power. The tested "alien" contention behaviors and the parameters that represent the results are common to the trigger packet assessment.



(a) Master, BeeMon and CAOS (b) Slaves, Event Loggers and Processor (c) Slave, Event Logger and Processor

Figure 6.10: Optimized implementation: WFTT testbed assessment photos

Figure 6.10 presents multiple photos of the experimental setup used to assess the WFTT slave transmission timeliness. Figure 6.10(a) illustrates the master station, the BeeMon device and the CAOS tool. The WFTT slave setup and the corresponding WITAS elements (Event Loggers and Processor) are pictured over a PVC table in Figure 6.10(b). Figure 6.10(c) provides close-up photos of the two WITAS elements used in the WFTT timeliness assessment. At the top, a photo of the Event Logger (EL2) attached to its designated slave station (S2) is shown. At the bottom, a lateral photo of the Event Processor is presented, illustrating a uMRF board attached to the specific Event Processor interfacing board.

6.3.2 Results

This section presents and discusses the results obtained using the aforementioned testbed. In the following subsections, a comparative analysis between the timeliness of the different types of traffic supported by the WFTT protocol is provided.

Real-Time Transmissions

The results of the real-time transmissions are presented in Table 6.8. As introduced in the methodology section, a total of 1000 real-time transmissions were expected to be logged per trial in response to the broadcast of 167 trigger packets. This meant that, for

Table 6.8: Optimized WFTT real-time transmission timeliness

PNS	Contender Type	Contender Position	SUCC (#)	FAIL T ≀ Z (#)	RPER (%)	TPER (%)	MIN (μs)	MAX (μs)	AVG (μs)	STDV (μs)
BB	None	—	1000	0 ≀ 0	0.0	0.0	1531	1550	1541.7	4.55
	Wi-Fi	@-3m	160	164 ≀ 106	50.6	40.7	1534	1550	1541.7	4.58
		@+6m	41	7 ≀ 0	14.6	95.2	1533	1550	1540.1	5.17
	802.15.4	@-3m	1000	0 ≀ 0	0.0	0.0	1530	1550	1541.9	4.59
		@+6m	999	1 ≀ 0	0.1	0.0	1531	1550	1542.0	4.80
PI	None	—	999	1 ≀ 0	0.1	0.0	1531	1550	1541.6	4.80
	Wi-Fi	@-3m	1000	0 ≀ 0	0.0	0.0	1534	1550	1541.7	4.52
		@+6m	887	41 ≀ 0	4.4	7.2	1530	1550	1542.2	4.60
	802.15.4	@-3m	624	28 ≀ 28	4.3	34.7	1534	1550	1541.6	4.60
		@+6m	669	37 ≀ 28	5.2	29.3	1534	1550	1542.3	4.71

SUCC	*Succeeded Transmissions*	AVG	*Average Delay*
FAIL	*Failed Transmissions*	STDV	*Delay Standard Deviation*
RPER	*Real-Time Packet Error Rate*	T ≀ Z	[T]otal ≀ [Z]ero Receptions
TPER	*Trigger Packet Error Rate*	BB	*Black-burst Interference*
MIN	*Minimum Delay*	PI	*Protective Interference*
MAX	*Maximum Delay*		

each trigger packet, a total of six real-time transmissions were due to be logged. Provided that there were three slave stations, that each station performed a real-time transmission in its designated slot and that this transmission was logged by the two remaining slave stations, a total of six reception events should be recorded per elementary cycle.

Regarding Table 6.8 parameters, the real-time packet error rate (RPER) corresponds to the percentage of unsuccessful real-time packet transmissions. As before, if a reception event resulting from a real-time packet was generated, that transmission was said to be successful. The real-time packet error rate (RPER) matches the total (T) number of failed transmissions (FAIL) divided by the total number of possible logged transmissions (1000). The total number of failed transmissions resulting from the simultaneous absence of real-time packet receptions in both listening slave stations is represented in the table's Z column. The TPER stands for the trigger packet error rate and it reports the percentage of trigger packets that were not received by any of the slave stations. Hence, if a given trigger packet was received by at least one slave station, its transmission was considered successful.

Trials secured by BB interference

The trials secured using *black-burst* (BB) interference encompassed the evaluation of the WFTT protocol timeliness in three different scenarios of "alien" noise contention: no noise, Wi-Fi noise and IEEE 802.15.4 noise. The following subsections discuss the obtained results for each one of these scenarios.

Noise-free scenario

As depicted in Table 6.6, there were no packet failures in the noise-free case, as expected. When compared to the real-time packet delay values estimated in Table 4.6 (Section 4.4), the variation in percentage is small, of 3.1 % and 2.0 % for the minimum and maximum delays, respectively. Since the referred real-time packet delay estimates were determined using approximate values obtained in the *turbo mode* operation, the observed difference may result from that option.

Figure 6.11 depicts the delay histograms of three key trials. The first one (no contender) corresponds to the case where the WFTT network was secured by *black-burst* interference, but operated without contention from "alien" technologies. The second and third trials coincide with those having the highest RPER/TPER combination while employing either *black-burst* or protective interference to guard the WFTT network. As it will be seen further ahead, these delay histograms allow establishing a direct comparison between the different trials in terms of delay dispersion. From Figure 6.11, it is possible to observe that the delay value scattering for the "no contender" trial does not seem to follow a specific distribution.

Wi-Fi "alien" immunity

As introduced, the WFTT network was exposed to Wi-Fi noise generated by a contender placed at either at -3 or at +6 meters from the master. In the first case, the measured RPER was of 50.6 % with a TPER of 40.7 %. The lost packets resulting from zero receptions represented 64.6 % of the total number of lost packets. When compared to the noise-free case, the associated delay parameters exhibited a maximum variation of only 0.66 % (standard deviation). In the second case, the experienced RPER was of 14.6 % and the TPER was of 92.5 %. In this case the delay parameters suffered a maximum aggravation of 13.6 % (standard deviation).

These results indicate a significant aggravation of the TPER when the contender is moved from the -3 to the +6 meter position, following a similar trend evidenced in the trigger packet assessment. This trend indicates that the TPER increases with the distance

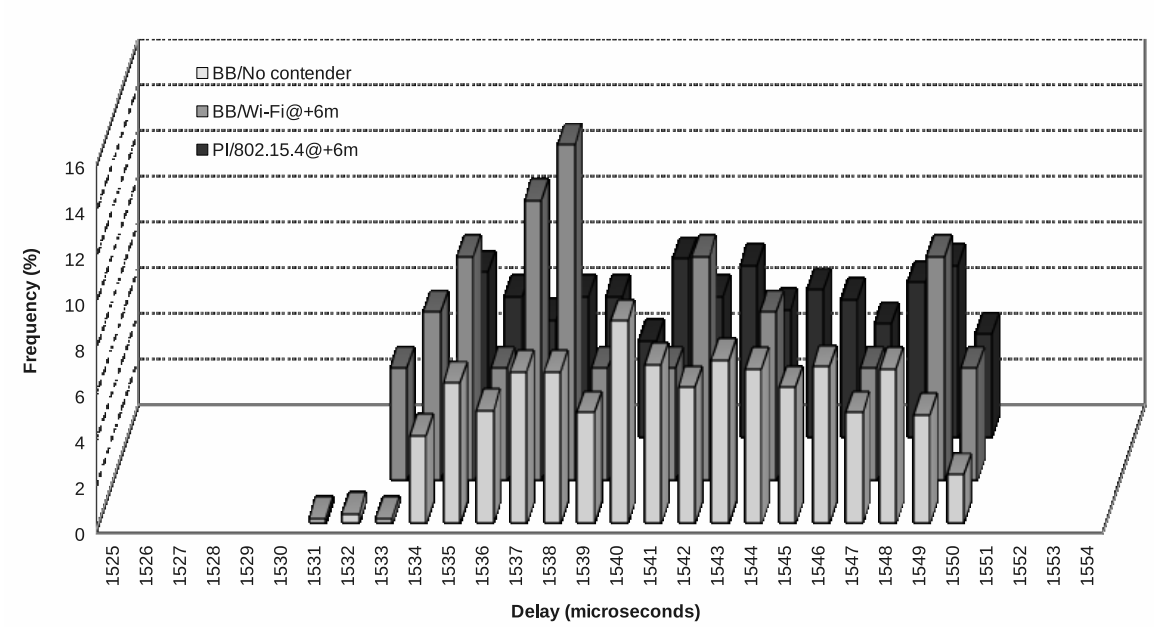


Figure 6.11: WFTT real-time packet transmission histograms

from the contender to the master station. With regards to trigger packet assessment, although this behavior is mainly caused by the existence of idle intervals that allow "alien" stations to perceive the medium free and initiate transmissions that ultimately corrupt the trigger packet, it is aggravated by the reduction of power perceived by the contender, as a result of increasing the distance to the master's PNS.

Regarding the RPER, a noticeable decrease is documented when the contender is moved from the -3 meters to the + 6 meters position. The overall number of packets used to calculate the RPER, however, decreases drastically, which limits the analysis and justification of this change. The RPER obtained in the -3 meter position was very high and results from a significant number of successful trigger packet transmissions. The optimized implementation of the WFTT protocol encompassed a 100 microsecond time slack between transmissions on the protected window. As argued, this interval was included to cope with the response time limitations of both the MRF24J40 transceiver and of the PNS. However, despite being significantly smaller than the one used in the unoptimized version (400 microseconds), it was still much larger than the Wi-Fi's minimum SIFS (10 microseconds), thus still resulting in a very high number of lost real-time packets.

The recorded delay parameters show a small variation for both contender positions when compared to the noise-free case. The exception occurs for the standard deviation delay in the +6 meter position. In this case, there was a higher dispersion around the average delay

value, possibly caused by the reduced number of samples. This dispersion can be observed in Figure 6.11, where the delays seem to exhibit a more randomized frequency distribution in comparison to the noise-free case.

IEEE 802.15.4 "alien" immunity

The WFTT network was subjected to IEEE 802.15.4 "alien" noise from a contender placed in two distinct positions: at -3 meters or at +6 meters from the master. As documented in Table 6.8, both cases depict a negligible TPER and RPER, in addition to the non significant variations in the delay parameters, when compared to the reference trial (noise-free case). The exception is the standard deviation, which shows a maximum variation of 5.5 %. However, because it is a measure of dispersion, this variation only indicates a slightly higher scattering of delays around the average value.

The negligible TPER and RPER results are a strong indicator of the effectiveness of the WFTT protocol in supporting the reliable transmission of trigger and real-time packets in environments affected by IEEE 802.15.4 "alien" noise. However, as noticed in the trigger packet timeliness assessment, the contender offers a low contention duty cycle, which possibility contributes to the almost inexistent loss of packet. In order to better characterize the WFTT protocol immunity to IEEE 802.15.4 based noise, a harsher environment should be used.

Overall, it is reasonable to conclude that the optimized implementation of the WFTT protocol is capable of ensuring a high level of reliability and timeliness for real-time transmissions in the presence of IEEE 802.15.4 "alien" noise when such transmissions are secured by *black-burst* interference.

Trials secured by PI interference

The obtained results for each one of the three distinct protective interference (PI) secured trials of "alien" noise contention are presented and discussed in the following subsections.

Noise-free scenario

As aforementioned, the noise-free scenario sets the reference performance for the remaining trials. The option to conduct different reference assessments for the *black-burst* and protective interference scenarios was motivated by the need of checking if the WFTT protocol optimization was responsible for some variation in the result's timeliness. Consider, for example, the *black-burst* protected scenario. If the RPER increased in the noise-

free case when compared to the unoptimized implementation, it would indicate an overlapping between PNS interference and WFTT data transmissions, which could be caused by the use of a time slack not long enough to cope with the timing limitations of the PNS and of the MRF24J40 transceiver.

Table 6.8 reports a null TPER and a 0.1 % RPER resulting from one real-time packet failure. These results confirm that the optimized WFTT protocol implementation using protective interference is working correctly. This is further corroborated by the delay results, which are similar to those of the previous reference trial.

Wi-Fi "alien" immunity

The RPER and TPER for the scenario where the contender is placed at -3 meters from the master were null, while the delay parameters suffered a maximum variation of 5.8 % (standard deviation). A degradation of these results is observed when the contender is at the +6 meter position. In this case, the RPER and TPER increase to 4.4 % and 7.2 %, respectively. In the first case, it is possible to observe that the protective interference is capable of totally hindering the Wi-Fi "alien" stations from initiating transmissions. In the second case, a small percentage of trigger and real-time packets fail their correct transmission. The real-time packet failures never occur simultaneously in both receiving slaves, i.e., at least one of the slaves is able to successfully receive the real-time packet. These small packet error rates can be justified by the occurrence of "alien" transmissions during the capture interval (trigger packet errors) or during the idle intervals in the protected window slots (real-time packet errors). In both cases, the "alien" transmissions are initiated when a Wi-Fi network adapter has a request and finds the medium free for a given minimum amount of time. This scenario should never occur since the PNS synthesizes protective interference specifically to avoid such occurrences. However, because the CAOS employs multiple USB Wi-Fi network adapters with specific radiation and sensitivity patterns, it is possible that one of them has a sensitivity minimum in the direction of the master station. As a consequence, it can sporadically sense the medium with less energy than it actually has and, therefore, initiate the requested "alien" transmission, resulting in the corruption of a trigger packet.

The justification for the low RPER results is slightly different. Should it be the same, the real-time packet failures would be mainly due to "Zero Reception" occurrences, i.e., most of the packets would not be received by any of the slave stations. This is not the case, however. Results indicate that the experienced RPER is originated by packet failures in only one of the slave stations. Although the WITAS provides no information allowing to

check if these packet failures are systematically occurring at the same slave, it is reasonable to assume so. Hence, a possible justification for this occurrence is that, although an "alien" station finds sporadically the medium idle as a result of a sensitivity minimum in the master's direction, the subsequent "alien" transmission has a different impact on the slave stations, which are possibly receiving an ongoing real-time transmission. On one of them the ongoing transmission is corrupted and the slave is not able to successfully decode the packet. On the other, due to a favorable SNR, the packet is successfully decoded.

IEEE 802.15.4 "alien" immunity

The WFTT protocol "alien" immunity results show a significant TPER of 34.7 % and 29.3 % for both contender locations (-3 and +6 meters). The associated RPER is notably smaller, but still not negligible, reaching values of 4.3 % (-3 meter position) and 5.2 % (+6 meter position). In both cases, given the high percentage of "zero reception" errors, the majority of the packet failures occurred simultaneously in both receiver slave stations. As in the previous scenarios, the delay parameters remain almost unchanged when compared to the reference trial.

The high TPER can be attributed to the overlapping of the trigger packets with IEEE 802.15.4 "alien" transmissions. On account that the protective interference has no impact on blocking IEEE 802.15.4 "alien" stations from performing transmissions, since it is propagated on a different channel, it is possible that an "alien" packet may collide with an ongoing trigger packet transmission. Provided that the TPER specifically accounts trigger packet failures occurring at all receivers simultaneously, it is reasonable to assume that its high value is a consequence of packet corruption due to transmission overlapping. The low/ moderate RPER seems to have the same cause. However, because real-time packets are significantly shorter than the trigger packet, the collision probability is notably lower. The position of the contender seems not to have a high impact on the packet error rate and on the delay statistical values.

The delay histogram for the overall performance worst-case scenario is reported in Figure 6.11. The selected scenario corresponds to the contender placed at the +6 meter position. As it can be observed, the delay distribution has an arbitrarily form whose bounds are within the maximum delay variation of the reference trial. This indicates a high consistency of the delays among different contention environments.

Table 6.9: Optimized WFTT contention transmission timeliness

PNS	Contender Type	Contender Position	SUCC (#)	FAIL T ≀ Z (#)	CPER (%)	TPER (%)	MIN (μs)	MAX (μs)	AVG (μs)	STDV (μs)
PI	None	—	994	0 ≀ 0	0.0	0.6	1406	3662	2552.6	724.88
	Wi-Fi	@-3m	978	4 ≀ 2	0.4	1.2	1406	3662	2493.0	748.05
		@+6m	889	63 ≀ 4	6.6	4.8	1406	3662	2535.9	729.78
	802.15.4	@-3m	623	131 ≀ 130	17.4	24.6	1406	3662	2543.5	716.63
		@+6m	646	158 ≀ 112	19.7	19.8	1408	3662	2463.3	695.03

SUCC	*Succeeded Transmissions*	AVG	*Average Delay*
FAIL	*Failed Transmissions*	STDV	*Delay Standard Deviation*
CPER	*Contention Packet Error Rate*	T ≀ Z	[T]otal ≀ [Z]ero Receptions
TPER	*Trigger Packet Error Rate*	PI	*Protective Interference*
MIN	*Minimum Delay*	MAX	*Maximum Delay*

Contention Window Transmissions

Table 6.9 presents the contention window transmission timeliness results. In this evaluation scenario, the contention packet transmissions are only assessed for protective interference (PI). However, as before, the associated trials are conducted on three different "alien" contention environments: noise-free, Wi-Fi noise and IEEE 802.15.4 noise.

With the exception of the CPER, which accounts for the contention packet error rate, the parameters reported on Table 6.9 are similar to those presented in Table 6.8 and discussed in the corresponding section. The trial definitions (duration, number of trigger packets transmitted, etc.) are also identical. However, in this case, trials address the transmission of contention packets on the contention window instead of real-time packets on the protected window.

Noise-free scenario

The noise-free scenario refers to the case where there is no medium contention by "alien" stations. In this case, Table 6.9 shows that the TPER is extremely small (0.6 %) and the CPER is null. From the delay perspective, the variation in percentage when compared to the estimated minimum and maximum delays of Table 4.6 (Section 4.4) is 3.7 % and 0.9 %, respectively. Provided that the contention packet delay estimates are an approximation that employ timings obtained using the *turbo mode* operation, the variation seems acceptable.

Two delay histograms are provided in Figure 6.12. These histograms refer to the

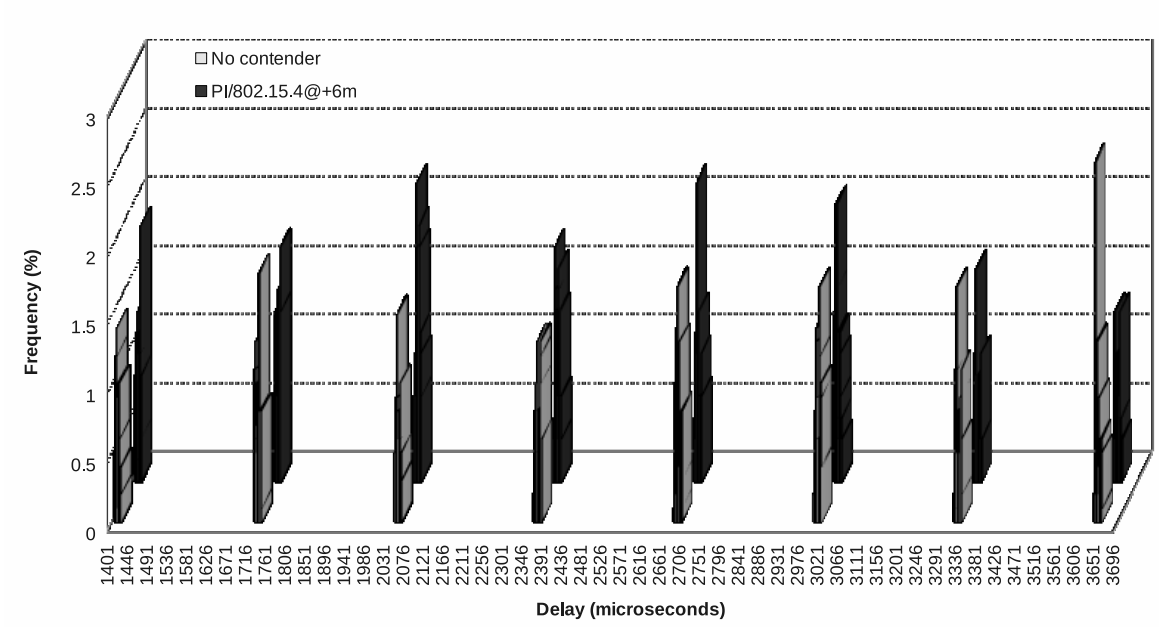


Figure 6.12: WFTT contention packet transmission histograms

best and worst transmission performance cases, as documented by Table 6.9. Hence, one corresponds to the case in which the WFTT network operates without contention from "alien" technologies and the other to the case in which the medium dispute is performed by an IEEE 802.15.4 "alien" contender. These two delay histograms allow a direct comparison between the best and worst case scenarios in terms of delay dispersion. Regarding the noise-free case, Figure 6.12 demonstrates that the delays concentrate in specific values, i.e., around 1414, 1734, 2056, 2373, 2692, 3013, 3333 and 3654 microseconds. These values were obtained from the histogram's raw data by selecting the central delay in each delay group. The average difference between these values is 320 microseconds with a standard deviation of 1.63 microseconds, which is consistent with the backoff period duration (320 microseconds) used as the time basis for the IEEE 802.15.4 CSMA/CA medium access algorithm. As detailed in Section 2.1.2, before initiating a CCA procedure, a random number of backoff periods ranging from 0 to $2^{BE} - 1$ must be enforced. Provided that the selected backoff exponent (BE) is 3, the random number of backoff periods to wait will range from 0 to 7, each one with a duration of 320 microseconds, which corroborates the results of Figure 6.12.

Wi-Fi "alien" immunity

The results reported in Table 6.9 document a TPER and CPER of 1.2 % and 0.4 %,

respectively, when the contender is at the -3 meters position. The TPER and CPER results are aggravated to values of 4.8 % and 6.6 %, if the contender is moved to the +6 meter position. In both cases, the minimum and maximum delays show no variation in comparison to the noise-free case. Conversely, the average delay and standard deviation suffer a maximum change of 2.3 %, thus indicating a slight difference in the delay distribution in the presence of Wi-Fi "alien" noise.

The TPER and CPER at the first position (-3 meters) are consistent with the Wi-Fi "alien" immunity results presented in the real-time packet timeliness evaluation scenario, i.e., the TPER and RPER values are very small when the medium is secured using protective interference. The results obtained at the +6 meter position also follow the observed increase trend in both packet error rates. It is worth noting that, as in the real-time packet evaluation scenario, the majority of the contention packet failures (93.6 %) are not originated by "zero reception" errors, which indicates that, at least, one of the slave stations is capable of successfully decoding the contention packet. As discussed above, the justification for these packet error rates is the occurrence of "alien" transmissions during the contention window (contention packet errors) or during the capture interval (trigger packet errors), which overlap with packets transmitted by the WFTT network. Since the contender (CAOS) employs several USB Wi-Fi network adapters characterized by different sensitivity patterns, it is possible that one of them has a sensitivity minimum in the direction of the master's PNS, thus causing it to sense the medium with less energy than it actually has. As a consequence, it will initiate the requested "alien" transmission, thus resulting in the corruption of an ongoing WFTT packet transmission.

In summary, the timeliness results of Table 6.9 indicate that the WFTT data transmissions performed within the contention window are highly secured against Wi-Fi "alien" interference.

IEEE 802.15.4 "alien" immunity

The IEEE 802.15.4 "alien" immunity results of Table 6.9 document a CPER of 17.4 % and 19.7 % when the contender is placed at either -3 or +6 meters from the master. In both cases, the percentage of packets lost due to "Zero Reception" errors is highly significant (99.2 % and 70.8%), which indicates that the packet transmission was logged, whereas the associated reception events were not. This is a consequence of the type of communications being performed. As explained, in contention-based IEEE 802.15.4 communications, a packet is only transmitted if the corresponding CCA procedure finds the medium idle. Otherwise, the access to the medium is postponed to a later instant and the process

repeats itself, up to a given maximum number of backoffs. If this number (typically four) is exhausted, then the CSMA/CA algorithm declares a channel access failure and no packet is transmitted. Since a packet is logged by the WITAS when its transmission is requested, a channel access failure will result in three "Zero Reception" errors. Hence, the significant CPER observed at both locations seems consistent with the occurrence of a high number of channel access failures. The increase of the CPER between positions also corroborates this conclusion, since the contender is closer to the WFTT slaves, thus exhibiting a higher level of energy and further aggravating the blocking effect.

Compared to the RPER results obtained in the analogous scenario, there is a notable increase in the packet error rate: 304.6 % and 278.8 % for the -3 and +6 meter positions, respectively. This accentuated increase can also be explained by the nature of the packets being transmitted. On the RPER case, the real-time packet was transmitted without sensing the medium. Therefore, if the SNR was favorable, the packet could be effectively received by one or more slave stations. On the CPER, the packet was only transmitted if the medium was found idle during the CCA procedure. This mechanism yields a higher packet error rate since, possibly, the packets that could benefit from a temporarily favorable SNR are potentially blocked by the transmitter itself and never get to be propagated in the medium.

Regarding the TPER documented in Table 6.9, a small reduction from 24.6 % to 19.8 % is visible when the contender is moved from the -3 to the +6 meter position. This relative error rate reduction (19.5 %) is similar to the one found in the analogous real-time packet assessment presented above (15.5 %). As concluded there, the observed errors seem to result from the overlapping of "alien" transmissions and trigger packets, which cause their corruption.

It is important to notice that the results presented above were obtained with an IEEE 802.15.4 "alien" contender characterized by a low transmission duty cycle. In this sense, in a more challenging contenting scenario, possibly including more than one contender, a degradation of the CPER should be expected.

Figure 6.12 demonstrates that the contention packet delays also concentrate around the specific values presented in the noise-free trial subsection. Furthermore, there is a visible difference in the delay dispersion pattern, which is consistent with the results presented in Table 6.9. Globally, the IEEE 802.15.4 "alien" contender noise has a small impact on the WFTT contention packet transmission delays.

Conclusions

The evaluation of the WFTT protocol slave transmission timeliness was conducted for two types of traffic. In the following subsections, the key conclusions for each one are summarized.

Real-time transmissions in the protected window

The Wi-Fi "alien" immunity results obtained for a WFTT network secured with *black-burst* interference indicated a high vulnerability to this type of interference. For example, the observed RPER was of 14.6 % for the +6 meter position and of 50.6 % in the -3 meter position. On the other hand, the resilience to IEEE 802.15.4 "alien" transmissions was found to be very high, thus supporting the conclusion that an optimized WFTT implementation, secured with *black-burst* interference, is capable of ensuring a high level of reliability for real-time transmissions in the presence of IEEE 802.15.4 "alien" noise.

It was observed that the optimized WFTT implementation was highly resilient to Wi-Fi noise when the master's PNS was configured to synthesize protective interference to secure the network. For example, even in a very harsh contention environment, encompassing multiple Wi-Fi adapters continuously disputing the medium, the RPER and TPER are bounded by values of 4.4 % and 7.2 %, respectively. Conversely, the results demonstrate that a WFTT network secured with protective interference has a significant vulnerability to IEEE 802.15.4 "alien" noise. These observations allow concluding that an optimized WFTT implementation employing protective interference is capable of supporting real-time transmissions with high reliability requirements in the presence of Wi-Fi noise.

According to the presented results, the delays are consistently bounded by a minimum and maximum values in the different reported scenarios of shielding interference and contending noise. This information establishes that, by combining both types of interference in the master's PNS, it is possible not only to support a high level of reliability against IEEE 802.15.4 and Wi-Fi "alien" noise, but also to guarantee a high level of determinism concerning the transmission delay.

Contention transmissions in the contention window

The timeliness results indicate that the WFTT contention data transmissions performed within the contention window are highly secured against Wi-Fi "alien" interference. For example, the TPER and CPER results are bounded by the 4.8 % and 6.6 % values, which

where obtained in the +6 meters position. Conversely, when the "alien" contender employs the IEEE 802.15.4 technology, the error rate is significantly aggravated. Therefore, the contention window protective interference is only capable of improving the reliability of contention packets when the "alien" source employs the Wi-Fi technology.

Regarding the timing behavior of the contention packets, it was found that the delays are concentrated around multiples of the backoff period duration (320 microseconds), which is the time basis for the IEEE 802.15.4 CSMA/CA medium access algorithm. In this sense, because the transceiver was configured with a backoff exponent of 3, the delays are concentrated around eight groups of delays in the range of 1406 to 3662 microseconds. This timing behavior supports the conclusion that the WFTT contention window is only capable of supporting best-effort traffic.

6.4 Summary

This chapter focused on providing an holistic experimental assessment of the WFTT protocol. This endeavor was segmented in three main evaluations. The first addressed an unoptimized version of the WFTT protocol where trials were conducted over a wide range of transmission power levels and "alien" noise contenders to collect information about the real-time/trigger packet error rate and the delay affecting these transmissions. This assessment allowed concluding that the time slack introduced to cope with the timing limitations of both the PNS and the MRF24J40MA transceiver devices was responsible for allowing "alien" stations to transmit in intervals reserved by the WFTT protocol for critical communications. Consequently, a reduction of the slack interval was proposed to minimize the error rates of both real-time and trigger packets.

Another conclusion from the first evaluation corresponded to the experimental validation that protective interference is highly effective in blocking Wi-Fi "alien" stations from accessing the medium and being able to initiate transmissions. Moreover, it was also concluded that a significant improvement in the timeliness of the WFTT protocol can be achieved when the interference and the data transmissions are propagated at their maximum power levels. Regarding the delay statistics for the real-time transmissions, results confirm highly consistent values across all trials and all noise/protection/power scenarios, thus suggesting that the WFTT protocol is suitable for supporting real-time communications.

The second evaluation scenario was built upon the conclusions of the first assessment.

In this (optimized) scenario, the scope of the evaluation was narrowed and focused on the trigger packet timeliness, considering the recommended transmission power levels of the previous evaluation. Therefore, the assessment was conducted using the maximum transmission power for both PNS interference and WFTT data. The two available types of PNS interference were tested against the two main technologies possibly contenting for the medium: Wi-Fi and IEEE 802.15.4. The goal of this assessment was to provide a first verification of the WFTT protocol optimized implementation.

The key conclusion arising from this study was that a combination of *black-burst* and protective interference is capable of ensuring a very high level of protection against both Wi-Fi and IEEE 802.15.4 "alien" noise in open environments. Furthermore, it was concluded that the minimum, maximum, average and standard deviation delays show no significant variation throughout the trials, regardless of the existence of noise and the type of noise used, which indicates that it is possible to ensure the transmission of trigger packets with a high level of determinism for both delay and error rate, even in challenging contention environments.

The third evaluation scenario was focused on the WFTT slave data transmission timeliness. This analysis has allowed to characterize the behavior of real-time packets, transmitted on the protected window, and of the contention packets sent on the contention window. Regarding the WFTT real-time packet timeliness, it was concluded that the *black-burst* interference is mostly effective in avoiding contention from IEEE 802.15.4 "alien" stations. Furthermore, it was found that the protective interference is highly effective in blocking Wi-Fi "alien" interference through evidence of highly reliable and timely communications. By combining both characteristics, it was concluded that reliable real-time communications can be supported by the WFTT protocol, as long as the PNS is capable of simultaneously synthesizing both *black-burst* and protective interference patterns.

According to the contention window packet transmission timeliness results, the protective interference propagated during this period offers a layer of shielding against noise cause by Wi-Fi "alien" stations. Hence, the transmission's delay characteristics are similar to the noise-free case in this scenario. However, the same does not occur for IEEE 802.15.4 "alien" stations because the protective interference has no influence on them.

"It is much better to do a little with certainty, and leave the rest for others that come after you, than to explain all things by conjecture without making sure of any thing."

Sir Isaac Newton (1643 - 1727)

7 Conclusions and Future Work

In the previous chapters, the Wireless Flexible Time-Triggered protocol was designed, implemented and experimentally validated. In the following sections, a short summary and discussion of the contributions of this dissertation are presented. Furthermore, some lines of interesting future research are also suggested.

7.1 Summary

The requirements of a communication protocol are the cornerstone of its design and development. In this sense, this dissertation began by addressing three representative application categories with emphasis on their communication requirements. The selection was driven by their adoption of personal area network technologies and operation in open spaces, where other technologies may contend for the medium. The study of these applications resulted in the conclusion that there is a broad heterogeneity of requirements, even within the same application domain, which poses demanding challenges regarding the flexibility of the core communication technologies and protocols. The key requirements of localization, monitoring and synchronization dependent applications can be resumed to a high dependability, resilience to (un)intended interference, security and flexibility in meeting different levels of timeliness, depending on the specific application being addressed. Hence, the design of an effective communication protocol must account for these requirements.

This dissertation argued that the use of a specific traffic separation mechanism, at the Medium Access Control (MAC) level, is the underlying foundation for designing a real-time communication protocol able to operate in "open communication environments". In order to validate this claim, research was conducted in several fields. A study of the background

subjects was focused on the most widespread wireless low-power technologies operating on the 2.4 GHz ISM band; on the main (low-power) wireless real-time protocols targeting factory automation applications for this band; and on the coexistence of the associated communication technologies.

Provided their widespread adoption for supporting communications on the 2.4 GHz ISM band, the Bluetooth and the IEEE 802.15.4 technologies were selected as the main candidates to enable the development of a wireless real-time communication solution. However, due to the emergent nature of the ANT and nanoNET technologies, these were also analyzed, although with less detail. The main conclusion is that, when compared to Bluetooth, the IEEE 802.15.4 protocol provides a much higher degree of flexibility, both in terms of its network architecture and timeliness support. The adoption of the IEEE 802.15.4 technology was also motivated by its use of the DSSS spread spectrum technique that can be combined with frequency agility mechanisms to avoid interference and provide reliable wireless communications in the 2.4 GHz ISM band. Despite these differentiating features, the IEEE 802.15.4 transceivers are amongst the most cost effective communication solutions in the market.

On account of its more demanding timeliness and reliability requirements, the study of the existing communication protocols for the 2.4 GHz ISM band was mainly focused on factory automation applications. The conducted analysis revealed that several standard wireless protocols (e.g., ISA SP100.11a, WirelessHART or WIA-PA) rely on the physical layer of the IEEE 802.15.4 standard for enabling communications. However, given their high complexity, multi-hop operation or particular configuration, significant end-to-end delays can potentially occur. Furthermore, there are protocols devised for specific purposes (e.g., the "Wireless Fieldbus for Plastic Machineries") which are also based on the IEEE 802.15.4 technology, but are able to meet more stringent timeliness requirements. This observation further validated the selection of the IEEE 802.15.4 technology as the underlying foundation of a new protocol. One aspect that emerged from this study was the common assumption that enough bandwidth is always available to support the protocol's operation. Despite the fact that some protocols employ frequency agility mechanisms, these are not effective when the available bandwidth is exhausted. In such scenarios, the studied technologies face a significant performance degradation, which may render them unusable. Therefore, to support applications with stringent reliability requirements, new communication methods were needed.

The study of the coexistence among communication technologies operating in the 2.4

GHz ISM band was focused on the IEEE 802.11, IEEE 802.15.4 and Bluetooth standards, given their massive widespread adoption. This study concluded that there is a significant degradation of the IEEE 802.15.4 packet reliability when its transmissions are exposed to noise from other co-located IEEE 802.15.4 stations. Likewise, it was found that the IEEE 802.11 networks are responsible for causing a dramatic reliability reduction in IEEE 802.15.4 networks. Conversely, the Bluetooth impact on IEEE 802.15.4 communications was considered negligible. This conclusion is supported by the Bluetooth use of a frequency-hopping scheme that, combined with a blacklisting mechanism, reduces the probability of collisions with neighbor technologies.

The channel access mechanisms reported in the literature are not suitable to meet the dependability and timeliness requirements of demanding real-time applications. Since all real-time protocols are highly dependent of the channel access timeliness, a novel technique named *bandjacking* was devised to provide deterministic wireless channel access in contention-based open environments, in particular, in the 2.4 GHz ISM band. This dissertation described, in detail, the architecture and operation of the stations developed to support this technique, besides reporting its implementation feasibility and discussing the results of its performance assessment. These results demonstrated that, although a commercial PNS can be developed using COTS components, two aspects must be accounted in any implementation. The first is the transceiver's latency to initiate/stop transmissions. The second is the PNS's delay to switch on/off the interference using SPI commands. In both cases, the delays were found to be significantly long and affected by a non-negligible jitter, which limited the determinism of the associated operations. Nevertheless, the *bandjacking* technique effectiveness was experimentally demonstrated with an implementation employing a COTS-based PNS.

Although the *bandjacking* technique was shown to support reliable wireless channel access in open environments, it was limited to transmissions from a real-time station to one (or more) standard station(s). In order to extend this determinism to a network of stations, requiring the support of multiple real-time data streams, a protocol named Wireless Flexible Time-Triggered (WFTT) was devised, building on the medium capture and maintenance conveyed by the *bandjacking* technique and inspired on the Flexible Time-Triggered paradigm. This dissertation reported its development in both architectural and operational perspectives. Moreover, two analytical studies focused on the protocol's implementation feasibility and timeliness were presented. The implementation feasibility was addressed considering the two identified problematic scenarios of "alien" contention:

Wi-Fi and IEEE 802.15.4 noise. In this sense, the timing bounds that the WFTT protocol must meet in both cases was calculated and justified. Regarding the protocol timeliness, the study was focused on both the synchronous and asynchronous traffic. In the first case, the delay and jitter associated to both trigger and real-time packet transmissions was studied. In the second case, the delay and jitter of a contention based transmission using the IEEE 802.15.4 protocol was presented.

The WFTT protocol instantiation in practical prototypes was also reported. The envisioned operation of the devices participating in a WFTT network was presented with an emphasis on their interaction and timing behavior. Furthermore, the architecture and inner operation of the master and slave stations was described. In this scope, their hardware/-software architectures and the state machines that rule their operation were documented. An aspect that was not considered in the implementation was the avoidance of idle periods in the protected window, which may occur when a slave station fails to transmit its scheduled packet. Nevertheless, one possible solution to mitigate this issue was proposed.

Finally, a setup for the WFTT protocol assessment was defined. Because open environments were characterized by encompassing a broad set of technologies; and since Wi-Fi was identified as one of the most pervasive wireless contention-based technologies, a tool allowing polluting the 2.4 GHz ISM band with Wi-Fi transmissions (CAOS) was devised. This tool is throughly used in the evaluation testbeds to mimic crowded environments. Furthermore, in order to evaluate the WFTT performance, a Wireless Timeliness Assessment System (WITAS) tool was also developed. This tool allows to individually measure the delay and jitter of multiple data flows. Besides the setup architecture, the protocol assessment described the methodology used to conduct the experimental trials, the obtained results and their analysis. The assessment was focused on providing an holistic evaluation of the WFTT protocol by isolating its behavior in three key evaluations scenarios. The first addressed an unoptimized protocol implementation, where trials were conducted over a wide range of transmission power levels and "alien" noise contenders to collect information about the real-time/trigger packet error rate and the delay affecting these transmissions. The second evaluation scenario inherited the improvement proposals emerging from the first scenario's analysis and narrowed the evaluation scope by focusing on the trigger packet timeliness of the WFTT optimized implementation. The third evaluation scenario was exclusively focused on the slave data transmission timeliness of this optimized implementation.

The collected results support the thesis that a specific traffic separation mechanism

(*bandjacking*) working at the MAC level is the design cornerstone of a communication protocol (WFTT) capable of supporting the requirements of real-time applications in open environments. Additionally, results demonstrate that it is feasible to (simultaneously) support flexibility and timeliness in environments encompassing multiple contention-based technologies. These conclusions are based on the observations that a combination of *black-burst* and protective interference is capable of ensuring a very high level of protection against both Wi-Fi and IEEE 802.15.4 "alien" noise transmissions in open environments; and that both WFTT trigger and real-time packets experience a high level of determinism regarding their delay and error rate, even in challenging contention environments. Furthermore, the existence of a contention window secured by protective interference offers an additional layer of shielding against noise cause by Wi-Fi "alien" stations, which enables the support of reliable best-effort low-power contention communications in environments encompassing Wi-Fi "alien" stations.

7.2 Future Research

This dissertation has focused on devising a wireless real-time medium access protocol that provides timeliness guarantees in open environments. Higher layer services such as, for example, admission control and message scheduling have been left out of the scope of this dissertation. Hence, one logical line of future research is the adaptation of existing FTT higher level mechanisms to the WFTT protocol. In addition, some extensions for improving the timeliness and dependability of the WFTT protocol, which seem promising, are the message/station redundancy support, and the multi-domain coordination and routing.

Scheduling and Admission Control Implementation

This dissertation proposed and validated a medium access protocol inspired by the FTT protocol for wireless communications. The assessment of the protocol was conducted using a set of messages with constant requirements and fixed scheduling. The flexibility potential provided by the WFTT protocol can only be fully exploited if an admission control mechanism and a scheduler are available, allowing modifying data flow requirements and adding/removing data streams on-line. Although the WFTT protocol is slightly different from the FTT, the implementation of the scheduling and admission control mechanisms should not be a difficult endeavor, since the WFTT master has more resources than the FTT-CAN master, for example, and their timing windows have a similar structure and

operational requirements.

Message/Station Redundancy Support

The WFTT protocol was proven to be an effective solution for supporting wireless real-time communications in contention-based environments. Although the most pervasive technologies operating on the 2.4 GHz ISM band rely on contention to access the medium, there are a few others (e.g., Bluetooth) which employ a TDMA MAC and could, eventually, compromise the WFTT timeliness. The same occurs for RF jammers. Therefore, the development of a redundancy mechanism that could replicate the WFTT protocol operation in more than one frequency band would improve the reliability of the protocol. Because the WFTT protocol relies on a single master to transmit a trigger message that triggers the operation of the slaves, if the master fails, the network becomes compromised. One interesting line of research is the implementation of a WFTT master replication mechanism that would improve the overall WFTT dependability.

Multi-domain Coordination and Routing

As presented in this dissertation, the support of large wireless networks is an important requirement in many application scenarios. Therefore, extending the WFTT master/multi-slave architecture to cope with this requirement is a compelling arena of future research. Two aspects must be addressed. First, different domains should be able to communicate with each other in a timely fashion. This involves devising mechanisms to ensure the routing of packets between WFTT networks (clusters) and, also, the phase coordination between them to guarantee that whenever a given WFTT master sends a packet to another neighbor master, the later is ready to receive it. Second, as each WFTT cluster operation is ruled by its master, a global scheduling paradigm could be devised to allow the reuse of the spectrum (frequency bands) and guarantee collision-free communications between clusters. One possible solution can be the adoption of a coloring mechanism similar to those employed in cellular radio communications.

Bibliography

[1] A. Flammini, P. Ferrari, D. Marioli, E. Sisinni, and A. Taroni, "Wired and Wireless Sensor Networks for Industrial Applications," *Microelectronics Journal*, vol. 40, no. 9, pp. 1322 – 1336, 2009, quality in Electronic Design; 2nd IEEE International Workshop on Advances in Sensors and Interfaces; Thermal Investigations of ICs and Systems.

[2] Ekahau, "Real-time Asset and People Tracking," April 2014. [Online]. Available: http://www.ekahau.com

[3] Exavera, "eShepherd," April 2014. [Online]. Available: http://www.exavera.com/

[4] Aeroscout, "Real-time Visibility," April 2014. [Online]. Available: http://www.aeroscout.com/

[5] Ubisense, "Precise Real-time Location," April 2014. [Online]. Available: http://www.ubisense.net

[6] J. Fonseca and P. Bartolomeu, "A MAC Protocol to Manage Communications in Localization Systems Based on IEEE802.15.4," in *Industrial Electronics, 2008. IECON 2008. 34th Annual Conference of IEEE*, 2008, pp. 2717 –2723.

[7] W.-C. Park and M.-H. Yoon, "The Implementation of Indoor Location System to Control ZigBee Home Network," *SICE-ICASE, 2006. International Joint Conference*, pp. 2158–2161, 18-21 Oct. 2006.

[8] S.-U. Yoon, L. Cheng, E. Ghazanfari, Z. Wang, X. Zhang, S. Pamukcu, and M. Suleiman, "Subsurface Monitoring Using Low Frequency Wireless Signal Networks," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on*, March 2012, pp. 443–446.

[9] K. Lorincz, D. J. Malan, T. R. F. Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G. Mainland, M. Welsh, and S. Moulton, "Sensor Networks for Emergency Response: Challenges and Opportunities," *IEEE Pervasive Computing*, vol. 3, no. 4, pp. 16–23, 2004.

[10] C. Fischer and H. Gellersen, "Location and Navigation Support for Emergency Responders: A Survey," *Pervasive Computing, IEEE*, vol. 9, no. 1, pp. 38 –47, 2010.

[11] J. Wilson, V. Bhargava, A. Redfern, and P. Wright, "A Wireless Sensor Network and Incident Command Interface for Urban Firefighting," in *Mobile and Ubiquitous Systems: Networking Services, 2007. MobiQuitous 2007. Fourth Annual International Conference on*, 2007, pp. 1 –7.

[12] A.-K. Chandra-Sekaran, A. Nwokafor, P. Johansson, K. Mueller-Glaser, and I. Krueger, "ZigBee Sensor Network for Patient Localization and Air Temperature Monitoring During Emergency Response to Crisis," in *Sensor Technologies and Applications, 2008. SENSORCOMM '08. Second International Conference on*, 2008, pp. 233 –238.

[13] N. Patwari, J. Ash, S. Kyperountas, A. Hero, R. Moses, and N. Correal, "Locating the Nodes: Cooperative Localization in Wireless Sensor Networks," *Signal Processing Magazine, IEEE*, vol. 22, no. 4, pp. 54–69, July 2005.

[14] M. Lowton, J. Brown, and J. Finney, "Finding NEMO: On the Accuracy of Inferring Location in IEEE 802.15.4 Networks," in *2nd International ACM Workshop on Real-World Wireless Sensor Networks (REALWSN '06)*. Uppsala, Sweden: ACM Press, June 2006.

[15] J. Graefenstein and M. Bouzouraa, "Robust Method for Outdoor Localization of a Mobile Robot Using Received Signal Strength in Low Power Wireless Networks," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, May 2008, pp. 33 –38.

[16] D. He and L.-X. Zhang, "The Water Quality Monitoring System Based on WSN," in *Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference on*, April 2012, pp. 3661–3664.

[17] J. Tian, H. Wu, and M. Gao, "Measurement and Control System of Sewage Treatment Based on Wireless Sensor Networks," in *Industrial Technology, 2008. ICIT 2008. IEEE International Conference on*, April 2008, pp. 1–4.

[18] Y. Xi, W. Yang, N. Yamauchi, Y. Miyazaki, N. Baba, and H. Ikeda, "Real-time Data Acquisition and Processing in a Miniature Wireless Monitoring System for Strawberry during Transportation," in *TENCON 2006. 2006 IEEE Region 10 Conference*, November 2006, pp. 1–4.

[19] L. Zheng, "ZigBee Wireless Sensor Network in Industrial Applications," *SICE-ICASE, 2006. International Joint Conference*, pp. 1067–1070, 18-21 Oct. 2006.

[20] K. Goh, S. Ong, Y. Joe, P. Kusolpalin, W. Moh, and K. Ling, "FPGA Based Wireless Sensor Node for Distributed Process Monitoring," in *Industrial Electronics and Applications (ICIEA), 2012 7th IEEE Conference on*, July 2012, pp. 1934–1939.

[21] L. Q. Zhuang, D. H. Zhang, and M. M. Wong, *Wireless Sensor Networks for Networked Manufacturing Systems, Factory Automation*. InTech, 2010, ch. 7, pp. 201 – 220.

[22] B. Gao, S. Xiong, and Z. Xu, "The Application of Wireless Sensor Networks in Machinery Fault Diagnosis," in *Machine Vision and Human-Machine Interface (MVHI), 2010 International Conference on*, April 2010, pp. 315–318.

[23] Y. Wan, L. Li, J. He, X. Zhang, and Q. Wang, "Anshan: Wireless Sensor Networks for Equipment Fault Diagnosis in the Process Industry," in *Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON '08. 5th Annual IEEE Communications Society Conference on*, June 2008, pp. 314–322.

[24] T. F. Budinger, "Biomonitoring with Wireless Communications," *Annual Review of Biomedical Engineering*, vol. 5, pp. 383–412, August 2003.

[25] B.-S. Lin, B.-S. Lin, N.-K. Chou, F.-C. Chong, and S.-J. Chen, "RTWPMS: A Real-Time Wireless Physiological Monitoring System," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 10, no. 4, pp. 647 –656, 2006.

[26] D. Buxi, T. Berset, M. Hijdra, M. Tutelaers, D. Geng, J. Hulzink, M. van Noorloos, I. Romero, T. Torfs, and N. Van Helleputte, "Wireless 3-lead ECG System with On-board Digital Signal Processing for Ambulatory Monitoring," in *Biomedical Circuits and Systems Conference (BioCAS), 2012 IEEE*, November 2012, pp. 308–311.

[27] N. Samanta, S. Dey, and C. RoyChaudhuri, "Multi Sensor Wireless System Optimized for Elderly Health Monitoring," in *Computing Communication Networking Technologies (ICCCNT), 2012 Third International Conference on*, July 2012, pp. 1–8.

[28] N. Samanta, A. Chanda, and C. RoyChaudhuri, "Optimized Multi Sensor Wireless System for Elderly Health Monitoring," in *Sensing Technology (ICST), 2012 Sixth International Conference on*, December 2012, pp. 151–156.

[29] Institute of Electrical and Electronics Engineers, "Draft Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," *IEEE Unapproved Draft Std P1588/D2.2, Mar 2008*, 2008.

[30] D. Wobschall and Y. Ma, "Synchronization of Wireless Sensor Networks Using a Modified IEEE 1588 Protocol," in *Precision Clock Synchronization for Measurement Control and Communication (ISPCS), 2010 International IEEE Symposium on*, 272010-oct.1 2010, pp. 67 –70.

[31] M. Beffa, C. Day, K. Elder, S. Kooper, and K. Rose, "Synchronization of Modules in a Wireless Array," United States of America Patent EP 2 520 124 A2, November 7, 2012. [Online]. Available: https://register.epo.org/espacenet/application?number=EP10841717

[32] J. Tian, M. Gao, and H. Zhou, "Multi-channel Seismic Data Synchronizing Acquisition System Based on Wireless Sensor Network," in *Networking, Sensing and Control, 2008. ICNSC 2008. IEEE International Conference on*, April 2008, pp. 1269–1272.

[33] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, "Clock Synchronization for Wireless Sensor Networks: A Survey," *Ad Hoc Networks*, vol. 3, no. 3, pp. 281 – 323, 2005.

[34] D. Macii, A. Ageev, and A. Somov, "Power Consumption Reduction in Wireless Sensor Networks Through Optimal Synchronization," in *Instrumentation and Measurement Technology Conference, 2009. I2MTC '09. IEEE*, May 2009, pp. 1346 –1351.

[35] R. Leidenfrost and W. Elmenreich, "Establishing Wireless Time-triggered Communication Using a Firefly Clock Synchronization Approach," in *Intelligent Solutions in Embedded Systems, 2008 International Workshop on*, 2008, pp. 1 –18.

[36] C. Rentel and T. Kunz, "A Mutual Network Synchronization Method for Wireless Ad Hoc and Sensor Networks," *Mobile Computing, IEEE Transactions on*, vol. 7, no. 5, pp. 633 –646, May 2008.

[37] G. Gaderer, P. Loschmidt, A. Nagy, R. Exel, and T. Sauter, "Localisation in Wireless Sensor Networks," in *Sensors, 2009 IEEE*, Oct 2009, pp. 1004–1009.

[38] M. Kuhn, M. Mahfouz, J. Turnmire, Y. Wang, and A. Fathy, "A Multi-tag Access Scheme for Indoor UWB Localization Systems Used in Medical Environments," in *Biomedical Wireless Technologies, Networks, and Sensing Systems (BioWireleSS), 2011 IEEE Topical Conference on*, January 2011, pp. 75–78.

[39] N. Baker, "ZigBee and Bluetooth Strengths and Weaknesses for Industrial Applications," *Computing & Control Engineering Journal*, vol. 16, no. 2, pp. 20–25, April-May 2005.

[40] I. Sommerville, *Software Engineering*, 9th ed. Harlow, England: Addison-Wesley, 2010.

[41] USA Department of Defense, *Quality Assurance Terms and Definitions*, USA Department of Defense Std. MIL-Std-109C, 1994.

[42] ——, *System Safety Program Requirements*, USA Department of Defense Std. MIL-Std-882C, 1993.

[43] IBM, *IBM Dictionary of Computing*, 10th ed., G. McDaniel, Ed. New York, NY, USA: McGraw-Hill, Inc., 1993.

[44] A. A. Ucla, A. Avizienis, J. claude Laprie, and B. Randell, "Fundamental Concepts of Dependability," LASS-CNRS, Research report 1145, April 2001.

[45] S. Behnke, A. Egorova, A. Gloye, R. Rojas, and M. Simon, "Predicting Away Robot Control Latency," in *RoboCup 2003: Robot Soccer World Cup VII*, ser. Lecture Notes in Computer Science, D. Polani, B. Browning, A. Bonarini, and K. Yoshida, Eds. Springer Berlin Heidelberg, 2004, vol. 3020, pp. 712–719. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-25940-4_70

[46] Institute of Electrical and Electronics Engineers, "IEEE Standard for Local and Metropolitan Area Networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)," *IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006)*, pp. 1–314, 2011.

[47] P. Bartolomeu, J. Fonseca, N. Rocha, and F. Basto, *Communications in Medical Applications, The Industrial Electronics Handbook*, 2nd ed. CRC Press, February 2011., vol. 2, ch. 29, pp. 1 – 16.

[48] M. Neuman, J. Picconnatto, and J. Roux, "A Wireless Radiotelemetry System for Monitoring Fetal Heart Rate and Intrauterine Pressure During Labor and Delivery," *Gynecologic Investigation*, vol. 1, no. 2, pp. 92–104, 1970.

[49] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync Protocol for Sensor Networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, ser. SenSys '03. New York, NY, USA: ACM, 2003, pp. 138–149. [Online]. Available: http://doi.acm.org/10.1145/958491.958508

[50] C. Zhang, M. Kuhn, B. Merkl, A. Fathy, and M. Mahfouz, "Real-Time Noncoherent UWB Positioning Radar With Millimeter Range Accuracy: Theory and Experiment," *Microwave Theory and Techniques, IEEE Transactions on*, vol. 58, no. 1, pp. 9–20, 2010.

[51] F. Sivrikaya and B. Yener, "Time Synchronization in Sensor Networks: A Survey," *Network, IEEE*, vol. 18, no. 4, pp. 45–50, July 2004.

[52] Bluetooth SIG, "Specification of the Bluetooth System Version 4.0 + HS," pp. 1–2302, June 2010.

[53] Z. Alliance, "Document 053474r13," *ZigBee Specification*, pp. i–508, December 2006.

[54] Z-Wave Alliance, April 2014. [Online]. Available: http://www.z-wavealliance.org

[55] EnOcean GmbH, April 2014. [Online]. Available: http://www.enocean.com

[56] Bluetooth Special Interest Group, April 2014. [Online]. Available: http://www.bluetooth.org

[57] Bluetooth SIG, "Specification of the Bluetooth System Version 1.1," February 2001.

[58] Institute of Electrical and Electronics Engineers, "IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks-Specific requirements - Part 15.1: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs) ," *IEEE Std 802.15.1-2002*, pp. 1–1169, June 2002.

[59] Bluetooth SIG, "Specification of the Bluetooth System Version 1.2," November 2003.

[60] Institute of Electrical and Electronics Engineers, "IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks-Specific Requirements - Part 15.1: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs)," *IEEE Std 802.15.1-2005 (Revision of IEEE Std 802.15.1-2002)*, pp. 1–600, June 2005.

[61] Bluetooth SIG, "Specification of the Bluetooth System Version 2.0 + ERD," November 2004.

[62] ——, "Specification of the Bluetooth System Version 2.1 + ERD," July 2007.

[63] ——, "Specification of the Bluetooth System Version 3.0 + HS," April 2009.

[64] Institute of Electrical and Electronics Engineers, "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," *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, pp. C1–1184, 12 2007.

[65] J. Padgette and K. Scarfone, "Guide to Bluetooth Security, Revision 1 (Draft). NIST Special Publication 800-121," National Institute of Standards and Technology, Tech. Rep., September 2011.

[66] S. Mare and D. Kotz, "Is Bluetooth the Right Technology for mHealth?" in *USENIX Workshop on Health Security (HealthSec)*, Aug. 2010. [Online]. Available: http://bit.ly/1iVRpoi

[67] Nordic Semiconductor, "Ultra Low Power Wireless Solutions," April 2012. [Online]. Available: http://www.nordicsemi.com

[68] Texas Instruments, April 2014. [Online]. Available: http://www.ti.com/

[69] CSR, April 2014. [Online]. Available: http://www.csr.com/

[70] EM Microelectronic - Marin SA, April 2014. [Online]. Available: http://www.emmicroelectronic.com

[71] Institute of Electrical and Electronics Engineers, "IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)," *IEEE Std 802.15.4-2003*, pp. 1–670, 2003.

[72] HART Communication Foundation, "WirelessHART Technical Data Sheet," December 2013. [Online]. Available: http://www.hartcomm.org

[73] Internet Engineering Task Force, "IPv6 over Low power WPAN," December 2013. [Online]. Available: http://datatracker.ietf.org/wg/6lowpan

[74] Institute of Electrical and Electronics Engineers, "IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems- Local and Metropolitan Area Networks- Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)," *IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003)*, pp. i–305, 2006.

[75] ——, "IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirement Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)," *IEEE Std 802.15.4a-2007 (Amendment to IEEE Std 802.15.4-2006)*, pp. 1–203, 2007.

[76] Freescale, April 2014. [Online]. Available: http://www.freescale.com/

[77] Atmel, April 2014. [Online]. Available: http://www.atmel.com/

[78] Microchip, April 2014. [Online]. Available: http://www.microchip.com

[79] Dynastream Innovations Inc., April 2014. [Online]. Available: http://www.thisisant.com/

[80] ——, April 2014. [Online]. Available: http://www.dynastream.com/

[81] Nanotron Technologies GmbH, "nanoNET Chirp-based Wireless Networks," Nanotron Technologies GmbH, Tech. Rep. NA-04-0000-0298-1.04, November 2007.

[82] ——, April 2014. [Online]. Available: http://www.nanotron.com

[83] T. Crenshaw, A. Tirumala, S. Hoke, and M. Caccamo, "A Robust Implicit Access Protocol for Real-time Wireless Collaboration," in *Real-Time Systems, 2005. (ECRTS 2005). Proceedings. 17th Euromicro Conference on*, 2005, pp. 177 – 186.

[84] S. Kumar, V. S. Raghavan, and J. Deng, "Medium Access Control Protocols for Ad-hoc Wireless Networks: A Survey," *Ad Hoc Networks*, vol. 4, no. 3, pp. 326 – 358, 2006.

[85] D. Dzung, C. Apneseth, J. Endresen, and J.-E. Frey, "Design and Implementation of a Real-time Wireless Sensor/Actuator Communication System," in *Emerging Technologies and Factory Automation, 2005. ETFA 2005. 10th IEEE Conference on*, vol. 2, 2005, pp. 10 pp. –442.

[86] G. Scheible, D. Dzung, J. Endresen, and J.-E. Frey, "Unplugged but Connected [Design and Implementation of a Truly Wireless Real-time Sensor/Actuator Interface]," *Industrial Electronics Magazine, IEEE*, vol. 1, no. 2, pp. 25–34, 2007.

[87] V. Phua, A. Datta, and R. Cardell-Oliver, "A TDMA-Based MAC Protocol for Industrial Wireless Sensor Network Applications using Link State Dependent Scheduling," in *Global Telecommunications Conference, 2006. GLOBECOM '06. IEEE*, 272006-dec.1 2006, pp. 1 –6.

[88] ——, "A TDMA-Based MAC Protocol for Industrial Wireless Sensor Network Applications using Link State Dependent Scheduling," University of Western Australia, Tech. Rep., February 2006.

[89] H. Korber, H. Wattar, and G. Scholl, "Modular Wireless Real-Time Sensor/Actuator Network for Factory Automation Applications," *Industrial Informatics, IEEE Transactions on*, vol. 3, no. 2, pp. 111–119, May 2007.

[90] N. Pereira, B. Andersson, and E. Tovar, "WiDom: A Dominance Protocol for Wireless Medium Access," *Industrial Informatics, IEEE Transactions on*, vol. 3, no. 2, pp. 120–130, May 2007.

[91] A. Rowe, R. Mangharam, and R. Rajkumar, "RT-Link: A Time-Synchronized Link Protocol for Energy Constrained Multi-hop Wireless Networks," Carnegie Mellon University, Department of Electrical and Computer Engineering, Tech. Rep., August 2005.

[92] ——, "RT-Link: A Global Time-synchronized Link Protocol for Sensor Networks," *Ad Hoc Netw.*, vol. 6, no. 8, pp. 1201–1220, 2008.

[93] A. Flammini, D. Marioli, E. Sisinni, and A. Taroni, "Design and Implementation of a Wireless Fieldbus for Plastic Machineries," *Industrial Electronics, IEEE Transactions on*, vol. 56, no. 3, pp. 747 –755, march 2009.

[94] International Society of Automation, "ISA-100.11a-2009, Wireless Systems for Industrial Automation: Process Control and Related Applications." http://www.isa.org , June 2010. [Online]. Available: http://www.isa.org

[95] S. Y. Shin and F. Rezha, "Extending CAN Protocol with ISA100.11a Wireless Network," in *ICT Convergence (ICTC), 2012 International Conference on*, 2012, pp. 472–476.

[96] International Electrotechnical Commission, "IEC 62734/Ed.1: Industrial Communication Networks - Wireless Communication Network and Communication Profiles - ISA 100.11a," January 2014. [Online]. Available: http://www.iec.ch/

[97] R. Wagner and R. Barton, "Performance Comparison of Wireless Sensor Network Standard Protocols in an Aerospace Environment: ISA100.11a and ZigBee Pro," in *Aerospace Conference, 2012 IEEE*, 2012, pp. 1–14.

[98] International Electrotechnical Commission, "WirelessHART," *Std., Rev. IEC 62591 Ed1.0*, 2010.

[99] HART Communication Foundation, April 2014. [Online]. Available: http://www.hartcomm.org

[100] J. Song, S. Han, A. Mok, D. Chen, M. Lucas, M. Nixon, and W. Pratt, "WirelessHART: Applying Wireless Technology in Real-Time Industrial Process Control," in *RTAS '08: Proceedings of the 2008 IEEE Real-Time and Embedded Technology and Applications Symposium*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 377–386.

[101] P. Ferrari, A. Flammini, S. Rinaldi, and E. Sisinni, "Performance Assessment of a WirelessHART Network in a Real-world Testbed," in *Instrumentation and Measurement Technology Conference (I2MTC), 2012 IEEE International*, 2012, pp. 953–957.

[102] S. Petersen and S. Carlsen, "Performance Evaluation of WirelessHART for Factory Automation," in *Emerging Technologies Factory Automation, 2009. ETFA 2009. IEEE Conference on*, 2009, pp. 1 –9.

[103] International Electrotechnical Commission, "Industrial Communication Network—Fieldbus Specifications—WIA-PA Communication Network and Communication Profile," *IEC 62601 Ed 2.0*, 2014.

[104] T. Zhong, C. Mengjin, Z. Peng, and W. Hong, "Real-time Communication in WIA-PA Industrial Wireless Networks," in *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on*, vol. 2, 2010, pp. 600–605.

[105] A. S. Hornby, *Advanced Learner's Dictionary*, S. Wehmeier, Ed. Oxford University Press, 2000.

[106] J. Zheng and M. J. Lee, "A Comprehensive Performance Study of IEEE 802.15.4," in *Sensor Network Operations*. IEEE Press, Wiley Interscience, 2006, pp. 218–237.

[107] T. Sun, L.-J. Chen, C.-C. Han, G. Yang, and M. Gerla, "Measuring Effective Capacity of IEEE 802.15.4 Beaconless Mode," *Wireless Communications and Networking Conference, 2006. WCNC 2006. IEEE*, vol. 1, pp. 493–498, 2006.

[108] M. Bertocco, G. Gamba, and A. Sona, "Is CSMA/CA Really Efficient Against Interference in a Wireless Control System? An Experimental Answer," in *Emerging Technologies and Factory Automation, 2008. ETFA 2008. IEEE International Conference on*, 15-18 2008, pp. 885 –892.

[109] L. Lo Bello and E. Toscano, "Coexistence Issues of Multiple Co-Located IEEE 802.15.4/ZigBee Networks Running on Adjacent Radio Channels in Industrial Environments," *Industrial Informatics, IEEE Transactions on*, vol. 5, no. 2, pp. 157–167, 2009.

[110] N. Nordin and F. Dressler, "Effects and Implications of Beacon Collisions in Co-Located IEEE 802.15.4 Networks," in *Vehicular Technology Conference (VTC Fall), 2012 IEEE*, 2012, pp. 1–5.

[111] A. Sikora and V. Groza, "Coexistence of IEEE802.15.4 With Other Systems in the 2.4 GHz-ISM-Band," in *Instrumentation and Measurement Technology Conference, 2005. IMTC 2005. Proceedings of the IEEE*, vol. 3, May 2005, pp. 1786 –1791.

[112] S. Y. Shin, H. S. Park, S. Choi, and W. H. Kwon, "Packet Error Rate Analysis of ZigBee Under WLAN and Bluetooth Interferences," *Wireless Communications, IEEE Transactions on*, vol. 6, no. 8, pp. 2825 –2830, 2007.

[113] M. Herrera, A. Bonastre, and J. Capella, "Performance Study of Non-beaconed and Beacon-Enabled Modes in IEEE 802.15.4 Under Bluetooth Interference," in *Mobile Ubiquitous Computing, Systems, Services and Technologies, 2008. UBICOMM '08. The Second International Conference on*, 292008-oct.4 2008, pp. 144 –149.

[114] R. de Francisco, L. Huang, G. Dolmans, and H. de Groot, "Coexistence of ZigBee Wireless Sensor Networks and Bluetooth Inside a Vehicle," in *Personal, Indoor and Mobile Radio Communications, 2009 IEEE 20th International Symposium on*, 2009, pp. 2700–2704.

[115] R. Garroppo, L. Gazzarrini, S. Giordano, and L. Tavanti, "Experimental Assessment of the Coexistence of Wi-Fi, ZigBee, and Bluetooth Devices," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011 IEEE International Symposium on a*, 2011, pp. 1–9.

[116] S. Zacharias, T. Newe, S. O'Keeffe, and E. Lewis, "Coexistence Measurements and Analysis of IEEE 802.15.4 With Wi-Fi and Bluetooth for Vehicle Networks," in *ITS Telecommunications (ITST), 2012 12th International Conference on*, 2012, pp. 785–790.

[117] A. Lavric, V. Popa, I. Finis, A. Gaitan, and A. Petrariu, "Packet Error Rate Analysis of IEEE 802.15.4 Under 802.11g and Bluetooth Interferences," in *Communications (COMM), 2012 9th International Conference on*, 2012, pp. 259–262.

[118] W. Guo, W. Healy, and M. Zhou, "Impacts of 2.4-GHz ISM Band Interference on IEEE 802.15.4 Wireless Sensor Network Reliability in Buildings," *Instrumentation and Measurement, IEEE Transactions on*, vol. 61, no. 9, pp. 2533–2544, 2012.

[119] L. Angrisani, M. Bertocco, D. Fortin, and A. Sona, "Assessing Coexistence Problems of IEEE 802.11b and IEEE 802.15.4 Wireless Networks Through Cross-layer Measurements," *Instrumentation and Measurement Technology Conference Proceedings, 2007 IEEE*, pp. 1–6, 2007.

[120] K. Shuaib, M. Alnuaimi, M. Boulmalf, I. Jawhar, F. Sallabi, and A. Lakas, "Performance Evaluation of IEEE 802.15.4: Experimental and Simulation Results," *Journal of Communications*, vol. 2, no. 4, pp. 29–37, 2007.

[121] M. Petrova, L. Wu, P. Mahonen, and J. Riihijarvi, "Interference Measurements on Performance Degradation between Colocated IEEE 802.11g/n and IEEE 802.15.4 Networks," in *Networking, 2007. ICN '07. Sixth International Conference on*, 2007, pp. 93–93.

[122] R. Musaloiu-E and A. Terzis, "Minimising the Effect of WiFi Interference in 802.15.4 Wireless Sensor Networks," *International Journal of Sensor Networks*, vol. 3, no. 1, pp. 43–54, 2007.

[123] B. Polepalli, W. Xie, D. Thangaraja, M. Goyal, H. Hosseini, and Y. Bashir, "Impact of IEEE 802.11n Operation on IEEE 802.15.4 Operation," *Advanced Information Networking and Applications Workshops, International Conference on*, vol. 0, pp. 328–333, 2009.

[124] G. Yang and Y. Yu, "ZigBee Networks Performance Under WLAN 802.11b/g Interference," in *Wireless Pervasive Computing, 2009. ISWPC 2009. 4th International Symposium on*, 2009, pp. 1–4.

[125] P. Bartolomeu and J. Fonseca, "An Assessment of the IEEE 802.15.4 PHY Immunity to WiFi Interference," in *Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on*, sept. 2010, pp. 1 –4.

[126] J. W. Chong, H. Y. Hwang, C. Y. Jung, and D. K. Sung, "Analysis of Throughput in a ZigBee Network Under the Presence of WLAN Interference," *Communications and Information Technologies, 2007. ISCIT '07. International Symposium on*, pp. 1166–1170, 2007.

[127] S. Y. Shin, H. S. Park, and W. H. Kwon, "Mutual Interference Analysis of IEEE 802.15.4 and IEEE 802.11b," *Comput. Networks*, vol. 51, no. 12, pp. 3338–3353, 2007.

[128] Institute of Electrical and Electronics Engineers, "Supplement to 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: Higher-Speed Physical Layer Extension in the 2.4 GHz Band," *IEEE Std 802.11b-1999*, pp. i–90, 2000.

[129] ——, "IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems- Local and Metropolitan Area Networks- Specific Requirements Part Ii: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11g-2003 (Amendment to IEEE Std 802.11, 1999 Edn. (Reaff 2003) as amended by IEEE Stds 802.11a-1999, 802.11b-1999, 802.11b-1999/Cor 1-2001, and 802.11d-2001)*, pp. i–67, 2003.

[130] ——, "IEEE Standard for Information Technology-Local and Metropolitan Area Networks-Specific Requirements-Part 11: Wireless LAN Medium Access Control (MAC)and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput," *IEEE Std 802.11n-2009 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, and IEEE Std 802.11w-2009)*, pp. 1–565, 2009.

[131] J. Sobrinho and A. Krishnakumar, "EQuB-Ethernet Quality of Service Using Black Bursts," *Local Computer Networks, 1998. LCN '98. Proceedings., 23rd Annual Conference on*, pp. 286–296, Oct 1998.

[132] ——, "Quality-of-service in Ad-hoc Carrier Sense Multiple Access Wireless Networks," *Selected Areas in Communications, IEEE Journal on*, vol. 17, no. 8, pp. 1353–1368, Aug 1999.

[133] A. Lindgren, A. Almquist, and O. Schelen, "Evaluation of Quality of Service Schemes for IEEE 802.11 Wireless LANs," *Local Computer Networks, 2001. Proceedings. LCN 2001. 26th Annual IEEE Conference on*, pp. 348–351, 2001.

[134] L. Jacob, L. Xiang, and Z. Luying, "A MAC Protocol With QoS Guarantees for Real-time Traffics in Wireless LANs," *Proceedings of the Fourth International Conference on Information, Communications and Signal Processing*, vol. 3, pp. 1962–1966 vol.3, Dec. 2003.

[135] P. Bartolomeu and J. A. Fonseca, "Channel Capture in Noisy Wireless Contention-Based Communication Environments," in *Proc. of the 8th IEEE Int. Workshop on Factory Communication Systems*, France, May 2010.

[136] R. Svanbäck and D. Bolnick, "Intraspecific Competition Drives Increased Resource Use Diversity Within a Natural Population," *Proceedings of the Royal Society of London. Biological Sciences*, vol. 274, no. 1611, pp. 839–844, 2007.

[137] V. Geist, "The Evolution of Horn-Like Organs," *Behaviour*, vol. 27, no. 3/4, pp. 175–214, 1966.

[138] R. Moraes and F. Vasques, "A Probabilistic Analysis of Traffic Separation in Shared Ethernet Systems Using the h-BEB Collision Resolution Algorithm," in *13th International Conference on Real-Time Systems - RTS'2005*, 2005.

[139] National Instruments, "NI USRP-2922," April 2014. [Online]. Available: http://www.ni.com/usrp/

[140] Avnet, "ZYNQ SDR-II Eval," April 2014. [Online]. Available: http://www.zedboard.org/product/zynq-sdr-ii-eval

[141] Great Scott Gadgets, "HackRF - A Low Cost Software Radio Platform," April 2014. [Online]. Available: http://greatscottgadgets.com/hackrf/

[142] Ettus Research, LLC, "Universal Software Radio Peripheral," November 2013. [Online]. Available: http://www.ettus.com

[143] Firas Abbas Hamza, "The USRP under 1.5X Magnifying Lens," Tech. Rep., June 2008. [Online]. Available: http://gnuradio.org/redmine/attachments/129/USRP_Documentation.pdf

[144] Free Software Foundation, "GNU Radio," November 2013. [Online]. Available: http://gnuradio.org/

[145] P. Daponte, L. De Vito, S. Rapuano, C. De Dominicis, P. Ferrari, and A. Flammini, "Characterization of the A/D Conversion Section in Software Defined Radios," in *Instrumentation and Measurement Technology Conference (I2MTC), 2010 IEEE*, May 2010, pp. 357 –362.

[146] C. Boano, Z. He, Y. Li, T. Voigt, M. Zuniga, and A. Willig, "Controllable Radio Interference for Experimental and Testing Purposes in Wireless Sensor Networks," in *Local Computer Networks, 2009. LCN 2009. IEEE 34th Conference on*, 2009, pp. 865–872.

[147] D. G. Yoon, S. Y. Shin, W.-H. Kwon, and H. S. Park, "Packet Error Rate Analysis of IEEE 802.11b under IEEE 802.15.4 Interference," in *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*, vol. 3, 2006, pp. 1186–1190.

[148] Y. Mao, Z. Zhao, and X. Jia, "Understanding the Indoor Interference Between IEEE 802.15.4 and IEEE 802.11b/g Via Measurements," in *Wireless Communications and Signal Processing (WCSP), 2011 International Conference on*, 2011, pp. 1–5.

[149] T. Schmid, "GNU Radio 802.15. 4 En-and Decoding," NESL, Department of Electrical Engineering, University of California, Tech. Rep., 2005.

[150] A. Botta, A. Dainotti, and A. Pescapè, "A Tool for the Generation of Realistic Network Workload for Emerging Networking Scenarios," *Computer Networks*, vol. 56, no. 15, pp. 3531–3547, 2012.

[151] Wireshark Foundation, "Wireshark," April 2014. [Online]. Available: http://www.wireshark.org/

[152] Chipcon AS. (2007, March) CC2420 datasheet - 2.4 GHz IEEE 802.15.4 / ZigBee-Ready RF Transceiver (Rev. B).

[153] Microchip. (2010, August) MRF24J40 Data Sheet - IEEE 802.15.4 2.4 GHz RF Transceiver.

[154] ——. (2013, April) PIC32MX5XX/6XX/7XX Data Sheet - 32-bit Microcontrollers (up to 512 KB Flash and 128 KB SRAM) with Graphics Interface, USB, CAN, and Ethernet.

[155] ——. (2011, March) MRF24J40MC Data Sheet - 2.4 GHz IEEE Std. 802.15.4 RF Transceiver Module with PA/LNA and External Antenna Connector.

[156] ZigBee Alliance, "ZigBee - Control Your World," April 2014. [Online]. Available: http://www.zigbee.org

[157] Microchip, "MiWi Protocol," April 2014. [Online]. Available: www.microchip.com/miwi

[158] P. Bartolomeu, J. Fonseca, and F. Vasques, "Implementing the wireless FTT protocol: A feasibility analysis," in *Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on*, 2010, pp. 1–10.

[159] L. Almeida, P. Pedreiras, and J. Fonseca, "The FTT-CAN protocol: Why and How," *IEEE Transactions on Industrial Electronics*, vol. 49, no. 6, pp. 1189 – 1201, December 2002.

[160] FTT Paradigm, April 2014. [Online]. Available: http://paginas.fe.up.pt/~ftt/

[161] P. Pedreiras, L. Almeida, and P. Gai, "The FTT-ethernet Protocol: Merging Flexibility, Timeliness and Efficiency," *Real-Time Systems, 2002. Proceedings. 14th Euromicro Conference on*, pp. 134–142, 2002.

[162] R. Marau, L. Almeida, and P. Pedreiras, "Enhancing Real-Time Communication over COTS Ethernet switches," in *Factory Communication Systems, 2006 IEEE International Workshop on*, 2006, pp. 295–302.

[163] T. You, H. Hassanein, and C.-H. Yeh, *A Survey of Medium Access Control Protocols for Wireless Local and Ad Hoc Networks*, ser. Adaptation in Wireless Communications. CRC Press, 2009, ch. 2, pp. 39–83.

[164] L. Almeida, P. Pedreiras, J. Ferreira, J. Calha, J. A. Fonseca, R. Marau, R. Silva, and E. Martins, *Handbook of Real-Time and Embedded Systems*. Chapman and Hall/CRC, 2007, ch. Online QoS Adaptation With the Flexible Time-Triggered (FTT) Communication Paradigm, pp. 1–22.

[165] J. Ferreira, L. Almeida, and J. Fonseca, "Bus Guardians for CAN: A Taxonomy and a Comparative Study," in *Proceedings of the Latin-American Workshop on Dependable Automation System, satellite workshop of the Second Latin-America Symposium on Dependable Computing (LADC 2005)*, So Salvador da Baa, Brazil., October 2005, pp. 3–10.

[166] L. Boroumand, R. H. Khokhar, L. A. Bakhtiar, and M. Pourvahab, "A Review of Techniques to Resolve the Hidden Node Problem in Wireless Networks," *Smart Computing Review*, vol. 2, no. 2, pp. 95–110, 2012.

[167] Microchip. (2008, February) MRF24J40MA Data Sheet - 2.4 GHz IEEE Std. 802.15.4 RF Transceiver Module.

[168] LORCON, "Loss of Radio Connectivity library," April 2014. [Online]. Available: http://code.google.com/p/lorcon/

[169] Pedro Larbig, "RaLink RT73 USB Enhanced Driver," April 2014. [Online]. Available: http://homepages.tu-darmstadt.de/~p_larbig/wlan/

List of Acronyms

ACL Asynchronous Connectionless, 23
ADC Analog-to-Digital Converter, 96, 98, 175, A3
AES Advanced Encryption Standard, 110, 176
AFH Adaptative Frequency Hopping, 23, 26, 73
AM Amplitude Modulation, 53, 54
ANSI American National Standards Institute, 28
ARQ Automatic Retransmission Request, 43
ASK Amplitude Shift Keying, 28

BER Bit Error Rate, 77, 128
BLE Bluetooth Low Energy, 21, 22, 26, 27
BPM Burst Position Modulation, 28
BPSK Binary Phase-Shift Keying, 28
BR Basic Rate, 21, 22

CA Collision Avoidance, 32–34, 39, 56, 60, 61, 70, 76, 91, 110, 137, 140, 166, 168, 170, 175, 180, 186, 188, 191, 208, 218, 232, 234, 236
CAN Controller Area Network, 63, 134, 175, 176, 243
CAOS Contention-bAsed nOise Sequencer, ix, 17, 18, 126, 128, 129, 197, 201, 213, 214, 216–220, 224, 230, 234, 242, B3
CCA Clear Channel Assessment, 33, 34, 72, 74, 76, 78, 86, 88, 89, 91, 94, 126, 150, 155, 157, 162, 166, 167, 175, 183, 198, 209, 211, 218, 224, 232, 234, 235
CFP Contention Free Period, 92
COTS Commercial Off-The-Shelf, v, vi, 47, 64, 108, 109, 111, 113, 119–122, 124, 129, 130, 155, 156, 177, 241
CPLD Complex Programmable Logic Device, 95
CPU Central Processing Unit, 30, 174, 185
CSMA Carrier-Sense Multiple Access, 32–34, 39, 45, 56, 60, 61, 70, 72, 76, 87, 91, 110, 137, 140, 166, 168, 170, 175, 180, 186, 188, 191, 208, 218, 232, 234, 236

CSS Chirp Spread Spectrum, 28, 39

DAC Digital to Analog Converter, 92, 96, 98, 102
DCF Distributed Coordination Function, 91, 92
DFS Distributed Fair Scheduling, 84
DIFS Distributed Coordination Function Inter Frame Space, 92, 93
DMA Direct Memory Access, 174
DMIPS Dhrystone Millions of Instructions Per Second, 109
DoS Denial of Service, 23
DQPSK Differential Quadrature Phase-Shift Keying, 28
DSC Digital Signal Controller, 174, 175
DSG Digital Signal Generator, 91, 92
DSP Digital Signal Processor, 95
DSSS Direct Sequence Spread Spectrum, 23, 28, 67, 79, 91, 92, 156, 240

EC Elementary Cycle, ix, 139–145, 147, 148, 151, 152, 154, 164, 165, 167, 198, 200, 206, 214, 223, 224
EDCF Enhanced Distributed Coordination Function, 84, 85
EDR Enhanced Data Rate, 20–22, 73
EEPROM Electrically-Erasable Programmable Read-Only Memory, 185, A2
EIRP Equivalent Isotropically Radiated Power, 28, 92, 156
eSCO extended Synchronous Connection-Oriented, 23

FDD Frequency Division Duplex, 42
FEC Forward Error Correction, 23
FER Frame Error Rate, 77
FH Frequency Hopping, 42, 43
FHSS Frequency Hopping Spread Spectrum, 21
FPGA Field-Programmable Gate Array, 41, 95, 98, 103

FTT Flexible Time-Triggered, vi, 133–135, 138, 139, 166, 176, 243

GFSK Gaussian Frequency-Shift Keying, 21, 22, 28, 37
GTS Guaranteed Time Slots, 58, 69, 91, 92
GUI Graphical User Interface, A12

HART Highway Addressable Remote Transducer Protocol, 27
HMI Human-Machine Interface, 185, 186
HVAC Heating, Ventilation, and Air Conditioning, 3, 4, 10, 19

I^2C Inter-Integrated Circuit, 175
IEEE Institute of Electrical and Electronics Engineers, 28
IFS Inter-Frame Space, 34, 70, 84, 86, 88, 89, 94, 150, 155, 157, 172, 183, 192
ISM Industrial, Scientific and Medical, 16, 17, 19–21, 26–28, 37, 39, 40, 43, 47, 49, 55, 59–61, 64, 71, 79–81, 89, 91, 92, 96–98, 103, 108, 110, 115, 175, 239–243, B5
ISR Interrupt Service Routine, 120, 177, 187

LAN Local Area Network, 84
LED Light Emitting Diode, 186, A3
LQI Link Quality Indication, 77, 175
LSDS Link State Dependent Scheduling, 45

MAC Medium Access Control, 6, 16, 19, 21, 22, 27, 29, 32, 39, 40, 45, 56, 60, 62, 64, 69, 70, 80, 81, 83, 85, 87, 89, 129, 136, 159, 164, 175–177, 239, 242, 243
MCU Micro-Controller Unit, 38, 39, 109, 112–116, 120, 152, 174, 175, 177, 180, 185–187, A3
MDMA Multi Dimensional Multiple Access, 39
MIC Message Integrity Code, 30

MIMO Multiple-input Multiple-Output, 78
MIPS Millions of Instructions Per Second, 185, A2
MITM Man-in-the-middle, 23
MPDU Media Access Control Protocol Data Unit, 197, 213
MPSK M-ary Phase-Shift Keying, 28

O-QPSK Offset Quadrature Phase-Shift Keying, 28
OFDM Orthogonal Frequency-Division Multiplexing, 77
OSI Open Systems Interconnection, 37, 60, 80

PAN Personal Area Network, 30, 32, 34
PCF Point Coordination Function, 84
PDF Probability Density Function, 75, 76
PDR Packet Delivery Ratio, 77, 78
PER Packet Error Rate, 72–77, 81, 82, 101, 127–129, 194, 209, B7
PLCP Physical Layer Convergence Protocol, 197, 213
PLR Packet Loss Ratio, 77, 194
PNS Programmable Interference Synthesizer, v, ix, 95, 98, 99, 101, 103–117, 119–123, 125–130, 138, 140, 143, 144, 146, 150, 155, 156, 158–160, 162, 163, 167–170, 172, 176, 177, 180–184, 186, 193, 197, 200, 201, 204–211, 213, 214, 219–221, 227–230, 234, 236–238, 241
PSSS Parallel Sequence Spread Spectrum, 28
PWM Pulse Width Modulation, 175

QoS Quality of Service, 84, 85

RAM Random-Access Memory, 26, 35, 109, 174, 185, A2
RFF Radio Frequency Front-end, 96
RISC Reduced Instruction Set Computer, 185
RMS Root Mean Square, 14
RPM Revolutions Per Minute, 11

RSSI Receiver Signal Strength Indication, 2, 4, 6, 9, 77, 175
RTLS Real-Time Location Systems, 40

SCO Synchronous Connection-Oriented, 23
SDR Software Defined Radio, v, 95–98, 101, 103, 104, 107, 109, 129, 130
SFDR Spurious-Free Dynamic Range, 100
SIFS Short Inter-Frame Space, 211, 228
SIG Special Interest Group, 20–22
SIR Signal-to-Interference Ratio, 72, 77, 81, 82, 153, 196
SNR Signal-to-Noise Ratio, 76, 128, 203, 210, 220, 230, 235
SoC System-on-Chip, 26, 27
SPI Serial Peripheral Interface, 109, 110, 112, 116, 120, 121, 130, 160, 164, 167, 175–177, 180, 185, 186, 241
SWIG Simplified Wrapper and Interface Generator, 99

TDD Time Division Duplex, 21, 24
TDM Time Division Multiplex, 37
TDMA Time Division Multiple Access, 39, 41–43, 45–48, 53, 56, 57, 60, 61, 64, 65, 137, 140, 243
TDOA Time Difference of Arrival, 6
TP Trigger Packet, 139–141, 170–173, 181, 182, 189

UART Universal Asynchronous Receiver/ Transmitter, 109, 175–177, 186, A11
UDP User Datagram Protocol, 49, 104, 106
USB Universal Serial Bus, 99, 100, 102, 109, 197, 213, 230, 234, B6
USRP Universal Software Radio Peripheral, 95, 97–104, 106
UWB Ultra Wide Band, 28

VBR Variable Bit Rate, 85

WFTT Wireless Flexible Time-Triggered, vi, vii, ix, 17, 18, 137–144, 146, 150–155, 157, 160, 162–167, 169–173, 177–179, 183–189, 191–202, 204–214, 216–238, 241–244, A11, B1

WITAS WIreless Timeliness Assessment System, 17, 18, 123, 124, 171, 177, 179, 180, 186, 189, 192, 194–196, 199–201, 212, 213, 216, 218, 222, 224, 230, 234, 242, A1

WLAN Wireless Local Area Network, 2, 44, 67, 77, 78, 84

WPAN Wireless Personal Area Network, 1, 27, 92, 184

WSAN Wireless Sensor/Actuator Network, 47–49

WSN Wireless Sensor Network, 4, 19, 45, 77, 108

XML eXtensible Markup Language, 124

A

WITAS: A WIreless Timeliness Assessment System

This appendix describes the WITAS architecture, operation and supporting applications. This document is divided in four sections. The first corresponds to an overview of the system's architecture. Afterwards, an in-depth review of the protocol operation is presented with focus on both log and command activities. In this section, protocol examples are shown for clearness. The following section addresses the applications used to manage the WITAS system and to process the raw data collected in the trials. Finally, the implementation feasibility is studied and some lines of future improvement are analyzed.

A.1 Architecture

The WITAS architecture can be analyzed in two main perspectives: system and devices. The following subsections provide an architecture overview of the system and of the associated devices, with emphasis on their hardware.

A.1.1 System

The WITAS measurement system encompasses two types of devices: *Event Loggers* (ELs) and *Event Processors* (EPs), as shown in Figure A.1. The first ones are designed to be connected to the wireless stations (S_1 ... S_j) being assessed for their timeliness. Event Loggers register the transmission/reception instant of any packet sent/received by the associated wireless station. Additionally, ELs are designed to monitor the power consumption of the associated stations. All information registered by ELs is communicated

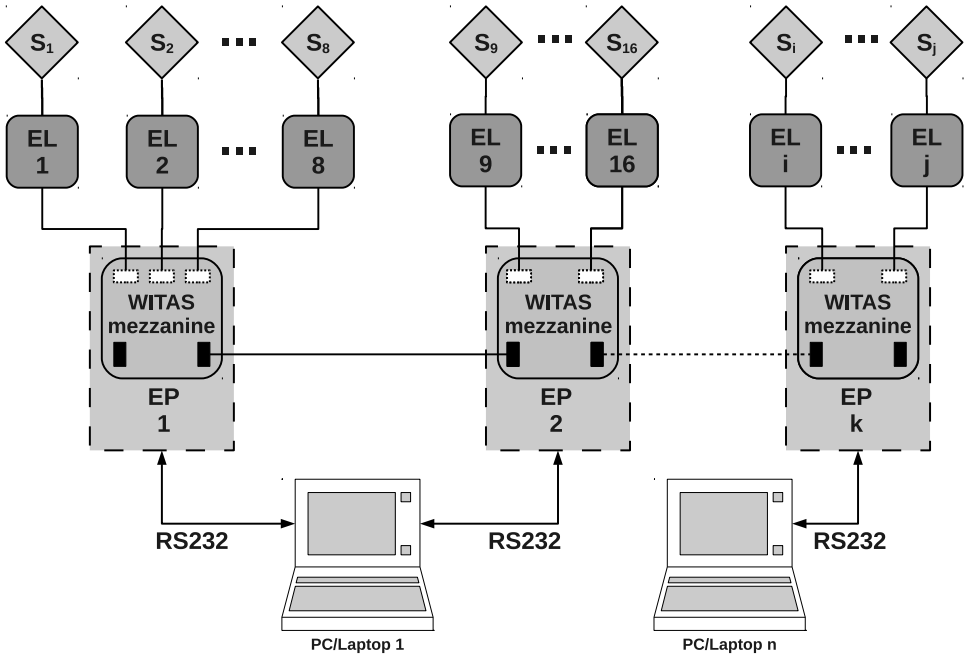


Figure A.1: WITAS global architecture

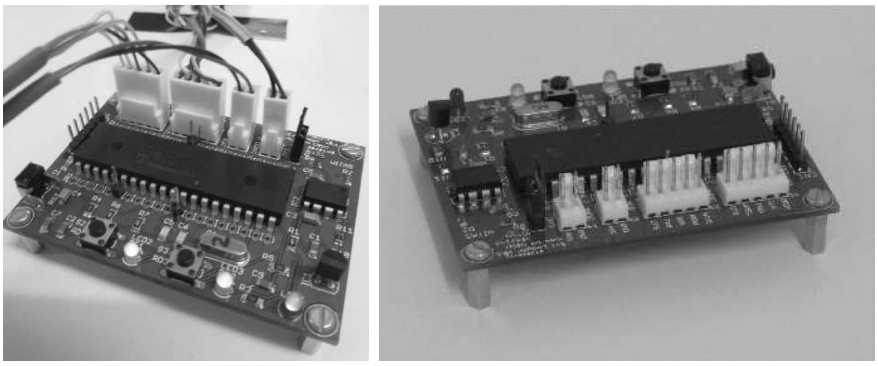
to the Event Processor, which is responsible for its forwarding to an application running in a PC.

The overall system behavior is driven by a central clock, which is supplied by one of the EPs to the remaining EPs and ELs. This shared clock allows that all time-stamping devices (Event Loggers) register the event instants in a coherent way, since the same clock is used by all Event Loggers.

Because all ELs are tied to a single serial reception port at the EP, the later needs to poll each of the associated Event Loggers individually to obtain the registered information. This is done in a round robin fashion by means of a specific serial command, which, despite being received by all ELs is only executed by the addressed EL. The ELs use a specific signal to notify the EP that event data sets are waiting to be collected. This allows speeding up the polling process and avoiding wasting bandwidth in queries that will have no response. In a similar fashion, EPs use a specific signal to drive ELs to a configuration mode.

A.1.2 Devices

As introduced, the WITAS system encompasses both ELs and EPs. The first are responsible for the event time-stamping while the second collect the event records from the ELs. The EL hardware architecture was specifically tailored to enable fast serial



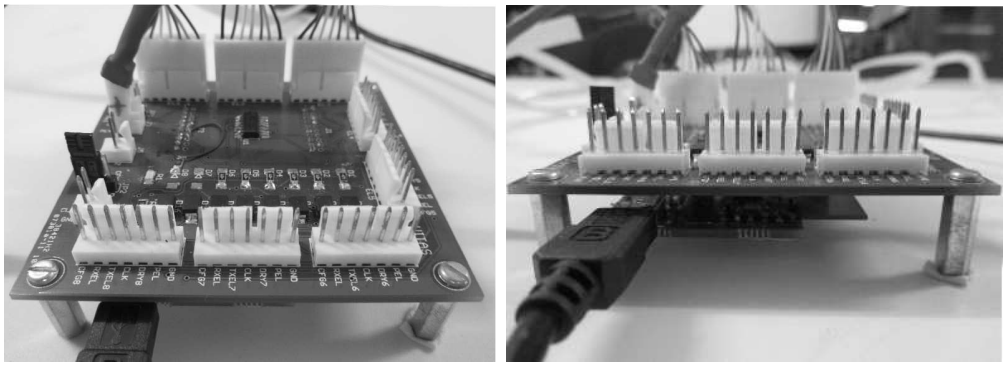
(a) Connected view (b) Unconnected view

Figure A.2: Event Logger pictures

communications with both the EP and the communication node being monitored. In this sense, the main attributes of the microcontroller selected to power the EL were the integration of two fast serial ports and a moderate/high processing capability. The selected microcontroller for this purpose was the Microchip PIC18F46K22 MCU, which can operate up to 16 MIPS (@64 MHz). Besides these features, the PIC18F46K22 provides a generous amount of memory (64 kilobytes of Flash, 3896 byte of RAM and 1024 bytes of EEPROM), four 16-bit hardware timers and 36 input/output pins.

Besides the PIC18F46K22 MCU, the Event Logger board encompasses an electronic circuit that powers-up the node being monitored. Jointly with the MCU's ADC, this circuit allows measuring the energy being supplied to the node. The board also includes several connectors, which allow feeding/receiving signals to/from external elements such as the EP or the communication node being monitored. Figure A.2 depicts two photos of an Event Logger. As documented, the board integrates several through-hole components. For example, the two LEDs allow checking the board's state while the two push buttons can be used to drive the EL operation.

The event processor is built using a uMRF and a mezzanine board. The uMRF board hardware architecture and features is detailed in Appendix C. The mezzanine board is simply an electrical interface that routes signals to/from the ELs from/to the uMRF board. The main signals being interfaced are the serial data (transmitted commands/received events), clock and data ready signal. An important design option of the mezzanine board was the direct connection of all the EL's transmit serial lines to the EP's receive pin. The EL's transmit pin is kept in a high impedance state, except for the intervals in which the EL performs a transmission. Moreover, another relevant EP design option was the connection



(a) Top view (b) Lateral view

Figure A.3: Event Processor pictures

of its transmit serial line to all the EL's receive pins. Therefore, the data transmitted by the EP is received simultaneously by all connected ELs. Provided that an EL only performs a data transmission to the EP when it is specifically addressed for this purpose, there is no risk of having overlapping transmissions from multiple ELs.

Figure A.3 illustrates two views of the Event Processor board. Figure A.3(a) shows a top view, where eight EL connectors are visible. Figure A.3(b) depicts a lateral photo registering the coupling between the mezzanine and the uMRF boards.

A.2 Operation

The operation of the WITAS measurement system is dependent of the behavior of three elements: wireless stations, Event Loggers and Event Processors. When the transmission of a packet is initiated, the corresponding wireless station switches the state of a digital line that is connected to an interrupt input of the Event Logger. As result, the EL records the instant at which the event occurred together with complementary information about the sender and message sequence number. This information is communicated by the wireless station (using the serial connection) immediately after signaling the digital line. An EL is capable of storing several event records before being queried by the EP to perform their transmission. When an EL has (at least) one event record to communicate, it raises a signal to the associated EP notifying this condition. Consequently, the EP will include this EL in the following event polling cycle, allowing the collection of the event records received so far. Hence, in the following polling cycle, the EL will be polled by means of a specific command to which it will reply with all the existing buffered event records (up

to a given configurable maximum). These records are then forwarded by the EP to an application running in a PC. Besides recording the serial stream of event records coming from the EP, the PC application allows configuring the WITAS operation (EPs and ELs) during the multiple phases of a trial. The features supported by this application are the following:

- Configure the trial parameters;
- Current sampling period by the EL[1];
- List of device IDs participating on the trial;
- Polling period of the EPs to the ELs;
- Stop, Set and Begin a trial;
- Record raw data file;
- Generate histogram CSV files;
- Store/load default settings.

The WITAS application is able to compute the following parameters based on the event records received from the EP:

- Latency (minimum, maximum, average, standard deviation);
- Latency histogram CSV file;
- Energy consumption (minimum, maximum, average, standard deviation)[1];
- Number of packets successfully transmitted;

A.2.1 Event Logging

The devised logging protocol was built to cope with the requirements of the WITAS monitoring system. Figure A.4 shows the structure of an event record frame. As documented, the frame start character '$' is followed by another character ('E', 'T' or 'R') defining the event's operation being logged. Afterwards, a character indicates the type of

[1]Although the EL hardware was designed to support current consumption, this feature is not yet supported by the EL's firmware.

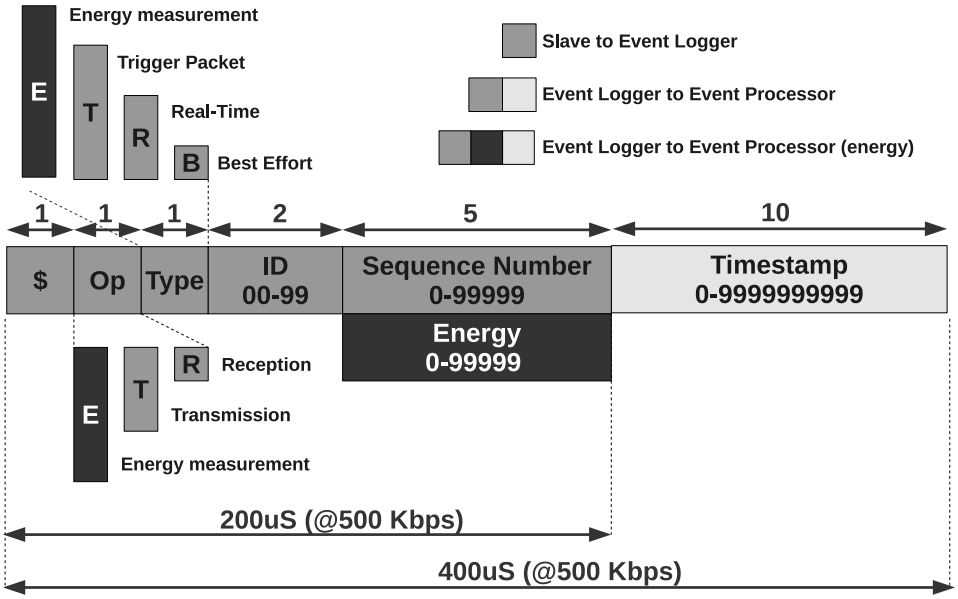


Figure A.4: Event logging frame format

packet registered (or 'E' if it corresponds to an energy measurement). Following, comes the identification of the station and the packet sequence number (or energy value). All of these fields are embedded in the frames sent by the wireless station to the EL. When the EL forwards the frame to the EP, it appends the timestamp field, whose value is represented in multiples of 1 microsecond. The baudrate employed in both serial communications is 500 kbps. Hence, the duration of the packet sent by the wireless station is 200 microseconds while its forwarding to the EP is twice as longer.

The sequence number together with the identification of the station allows tracking if any packets have been transmitted but not received by their intended recipients. Besides, because every event is time-stamped in a coherent manner, it is possible to compute the communication delays and other statistical parameters associated to the performance of the communication system being assessed.

Figure A.5 depicts two event logging examples. In the first, station "S00" indicates that it transmitted a trigger packet with the sequence number "00001". In the second, station "S01" informs that it received a real-time packet with the sequence number "00023" from station S04. When these packets are forwarded to the EP, the associated time-stamp is appended to the frame, as shown in Figure A.5. Afterwards, these packets are sent to the PC application, responsible for saving and processing the trial data.

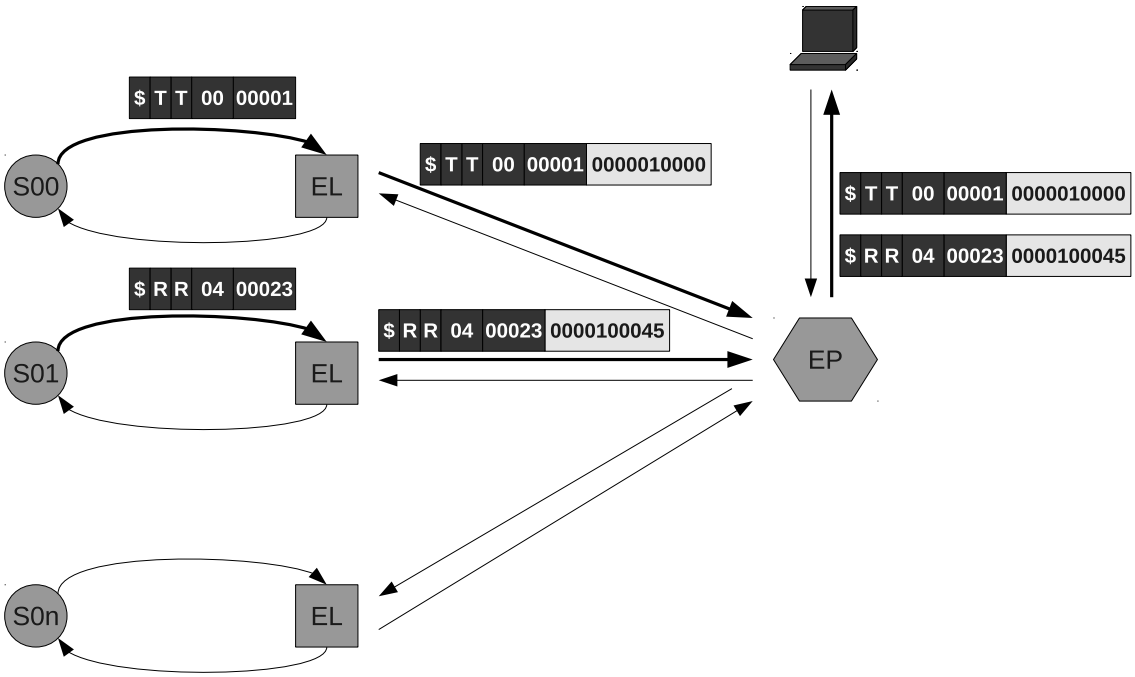


Figure A.5: Event logging example

A.2.2 Command and Control

Figure A.6 shows the WITAS control frame format used to configure and control the execution of measurement trials. As before, the frame started by the '$' initiator character, followed by the operation character and the identification of the device to be commanded. The frame length is typically constant with a size of 5 bytes. The exception is the configure command frame, whose size is variable and depends of the configuration payload. Except for the "get records" command, which is issued by the EP to the connected ELs, all commands are triggered by the PC application. The target device is specified in the "Dev" field of the frame together with its "ID". Regarding the identification of WITAS devices, the ELs are always configured with the ID of the associated wireless station. As for the EPs, the one with the zero ID is the "source", meaning that it is the one responsible to supply the clock for the timestamps.

Figure A.7 shows the PC application issuing two commands. In the first case, it requests that the Event Logger associated with the wireless station "S01" begins the logging procedure for upcoming events. In the second case, it instructs the wireless station "S00" to initiate its operation.

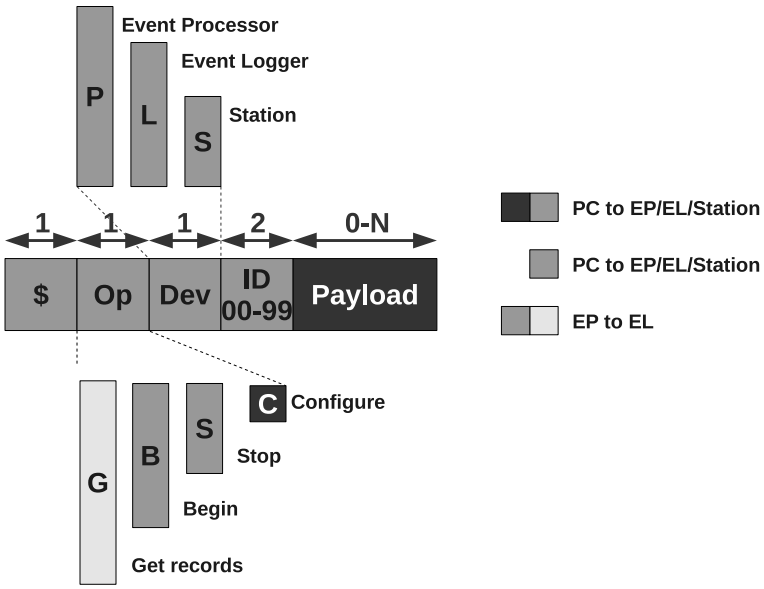


Figure A.6: WITAS control frame format

A.3 WITAS Applications

During a trial, the use of a PC application in combination with the WITAS system occurs in two phases. In the first, the application is required to configure/setup the target trial so that it carries on as expected. In the second, an application is needed to process the collected data and to obtain meaningful information from it. As introduced, all data exchanges in the WITAS system are conducted by means of serial messages. Hence, the selected applications must be able to perform serial data transmissions/receptions and analyze the information conveyed using this type of communication.

Docklight (www.docklight.de) is a popular tool for developing serial communication protocols, which, among other features, allows recording predefined sequences of symbols that can be sent via a serial port by simply pressing a button. In this scope, the Docklight PC application was used in combination with the WITAS system to setup the experimental trials described in this dissertation. The adopted approach was built upon the elaboration of a "command library" encompassing all the message sequences that should be transmitted to the WITAS system in order to properly configure the trials. After building this "library", the process of setting up a trial became very fast because the transmission of a given command only requires pressing the corresponding button in the Docklight application.

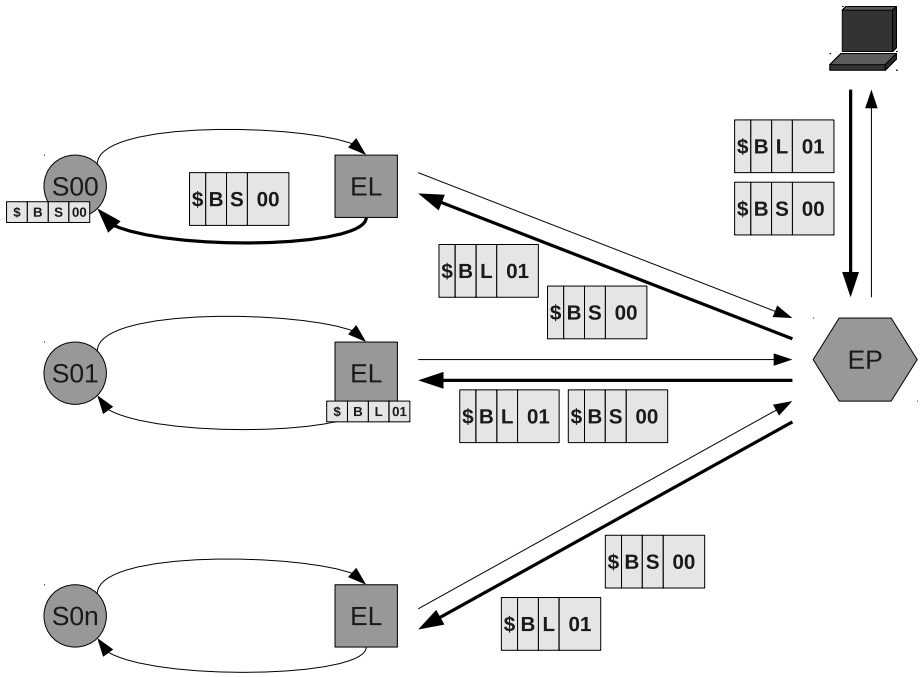


Figure A.7: WITAS control example

Listing A.1: Example of the results obtained by processing a trial's raw data

```
--> Transmission DELAY
    Minimum: 1406 us
    Maximum: 3662 us
    Average: 2598.8 us
    Std.Dev: 736.25 us

--> Transmitted PACKETS
    Success: 687
    Errors: 295
    ZR Errors: 4
    PER: 30.0

--> Received PACKETS
    Errors Total: 1
    Error Non-Repetitive: 1
```

A second PC application was required to process the trial's collected raw data. During a trial, the Docklight application receives all the messages forwarded by the EP. As described, these messages contain the EL event records plus their associated timestamps. Whenever a message is received, it is displayed in the Docklight's user interface. At the end of a trial, the received frames (plus timestamps) are manually copied to a text file, which can be used to obtain several statistical parameters. In this scope, a command-line application

was developed using the Ruby programming language. This application accepts the name of the text file where the messages are stored as an input parameter and reproduces the associated statistical results, as documented in Listing A.1. Optionally, the application can output a latency histogram using the comma separated value file format.

A.4 Implementation

This section is dedicated to the WITAS system feasibility analysis and to the discussion of the main limitations and improvements that can be performed in the future.

A.4.1 Feasibility Analysis

For the purpose of analyzing the feasibility of using the WITAS system to assess the WFTT protocol performance a typical elementary cycle (EC) of 200 milliseconds is assumed. An EC encompasses the transmission of a trigger packet (TP) followed by real-time (RT) and contention (CT) packets, if any. Considering a scenario encompassing one master, four RT stations and three contention-based stations performing one transmission each per elementary cycle, the events generated and sent to the Event Loggers will be:

- **Master:** 1 x TP transmission, 4 x RT reception, 3 x CT reception
- **Slave RT:** 1 x TP Reception, 1 x RT Transmission, 3 x RT Reception, 3 x CT reception
- **Slave CT:** 1 x TP Reception, 4 x RT Reception, 1 x CT Transmission, 2 x CT reception

Hence, the number of events reported in each EC by a wireless station to the associated Event Logger is given by:

$$\#EventsEC = \#RTPackets + \#CTPackets + 1 \quad \text{(A.1)}$$

Since each event takes 200 microseconds to be transmitted from the WFTT station to the Event Logger, the maximum number of packets that can be transmitted in an EC is given by:

$$\#EventsEC2EL = ECPeriod/200us \quad \text{(A.2)}$$

Regarding the forwarding of these events from the event logger to the Event Processor, the maximum number of packets that can be transmitted in an EC is given by:

$$\#EventsEC2EP = ECPeriod/400us \quad \text{(A.3)}$$

For a scenario where the EC period is 200 milliseconds, the #EventsEC2EL and #EventsEC2EP are characterized by values of 1000 and 500, respectively.

Considering the specified scenario, the transmission time of the events from the wireless stations to the Event Loggers is of 1.6 milliseconds, which corresponds to 0.8 % of the EC period. Furthermore, the transmission time of the event records stored at the Event Loggers during an EC is of 25.6 milliseconds, corresponding to 12.8 % of the available time in an EC. This interval corresponds to the delay of polling 64 event records, which are a result of the communications that occurred during the EC.

The scenario mentioned above defines a network encompassing eight WFTT wireless stations. This number is aligned with the limitation of an EP to collect event records from eight different Event Loggers. However, this number can be extended by using more EPs with additional Event Loggers. In this case, EPs can be connected in different serial ports at the PC or in different PCs running the logging application. In the first case, the event record stream is stored in a single file, which will be used to compute the communications' timeliness. In the second case, the event record stream is saved in multiple files. In this scenario, the partial record streams must be merged to obtain a complete view of the event history and allow computing the timeliness of the communications.

A.4.2 Limitations and future work

During the testing of the WITAS system and throughout the execution of the WFTT performance trials presented in this dissertation, several limitations were identified. Provided the system's support of the basic functionality, the correctness of the obtained results and the scarcity of time to redesign the hardware, the WFTT trials were conducted with the WITAS system in its current version. This version is characterized by some impairments, which, in most cases, result of early sub-optimal hardware design options that imposed constraints to the way the WITAS system can be effectively used. One of such limitations is the direct connection between UART interfaces, without employing appropriate drivers (e.g., EIA232 physical interfaces). Because of this, the distance between transmitter and receiver is severely shortened due to the signal attenuation and addition of noise. Hence,

the use of the WITAS system to assess networks extending over large physical areas is not currently possible. This effect is also experienced for other signals, namely the clock signal provided by the EP to the ELs. One possible solution for these problems can be the redesign of the EL and of the EP mezzanine boards in order to employ differential signals. The serial communications could employ a RS485 physical interface, for example.

One aspect that revealed to be a design drawback was the use of different architectures for the EL and for the EP. This option made the firmware development more complex and segmented. Another issue arising from this option was the EL UART's lack of flow control hardware support, which increased the difficulty of obtaining a consistent communication coordination between the EL and the EP. A possible solution for such problems could be the adoption of a common hardware architecture, employing the Microchip dsPIC33FJ256MC710 MCU. Besides being faster than the PIC18F46K22 used in the EL, it provides more memory (allowing buffering more event records) and integrates UARTs with flow control.

Although the trial setup process with the Docklight application is relatively fast, it forces the user to press a set of buttons in a (possibly) repetitive way. Furthermore, the processing and configure applications are decoupled from each other, which causes the user to individually setup each one. In order to overcome these drawbacks, a single configure and process GUI application could be built. This application should allow setting up a trial with a single click, record the raw data into a file and process it to obtain the designated statistical information. One important feature to include would be the possibility of saving/loading configuration files containing information about the different trial parameters (e.g., IDs of the stations, types of packets being considered, etc.). An alpha version of one such application was already implemented using the Java language.

B

CAOS: Contention-bAsed nOise Sequencer

CAOS stands for "Contention-bAsed Noise Sequencer" and, as suggested, it is a tool aimed at producing contention-based noise. The motivation for developing a device with this nature arose from the requirement of mimicking communication environments populated with Wi-Fi transmissions in the 2.4 GHz ISM band. Because the WFTT protocol was developed to operate over this band, its assessment in the presence of Wi-Fi contention noise was considered essential.

This appendix is divided in two main sections: the CAOS architecture and its evaluation. The first is focused on the employed hardware and on the software developed to support its functionality. The evaluation section addresses the CAOS effectiveness and is conducted using two different approaches. One in which its ability to "pollute" the 2.4 GHz ISM band is studied using a spectrum analyzer and the other where its effect on IEEE 802.15.4 broadcast transmissions is assessed.

B.1 Architecture

The CAOS architecture is very simple, thus indicating that a system with similar properties can be easily assembled. The following subsections describe the key elements of the CAOS hardware and software architectures.

B.1.1 Hardware

The hardware architecture of the CAOS encompasses a personal computer (PC) and a set of five USB Wi-Fi network adapters. The PC has no stringent requirements regarding processing power and memory. In this sense, a midrange PC setup was assembled using a

(a) Front (b) Back

Figure B.1: CAOS physical appearance

specific set of components, namely:

- Zotac H55-ITX motherboard;
- Intel Core i3-540 processor running at 3.06 GHz;
- 2GB RAM DIMM operating at 1333 MHz;
- Fujitsu 60 GB SATA 2.5 in HDD;
- Compucase 8K07 120 W Mini-ITX Case.

The PC was setup with a Mint Linux 10 operating system ("Julia" distribution, kernel 2.6.35) . Besides the PC components, a set of five BELKIN F5D7050 USB Wi-Fi network adapters were also employed in the construction of the CAOS. Figure B.1 depicts two photos of the CAOS: one from the front panel and the other from the back panel. As illustrated, provided that the Mini-ITX Case only has two front-facing USB ports, three USB extension cables were used to allow placing the Wi-Fi network adapters close to each other and with a consistent physical orientation.

B.1.2 Software

From the perspective of the user, the CAOS operation is managed through a web page, which is accessible by default at the (static) IP address 192.168.0.101. In this sense, if the user requires that Wi-Fi noise is generated, it must access the designated address using a

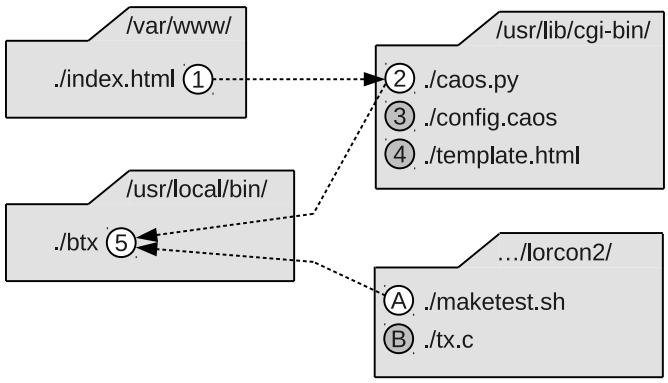


Figure B.2: CAOS software organization

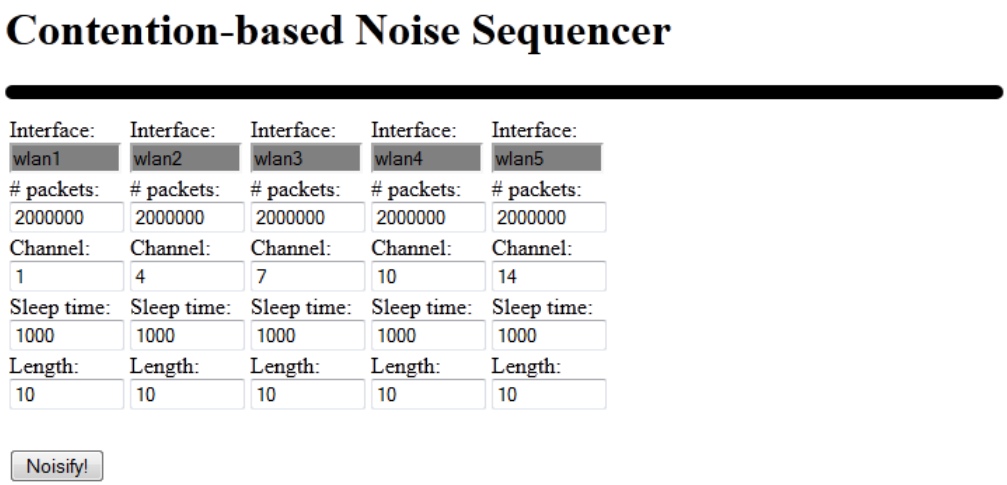


Figure B.3: CAOS on-line setup (browser snapshot)

browser and request the noise transmission to be initiated. In order to simplify the analysis of the software developed for the CAOS, a visual scheme was devised, showing its software organisation main elements. This scheme is represented in Figure B.2.

As documented, when the user accesses the 192.168.0.101 IP address using the browser, the server supplies the "index.html" file (1), which, in turn, redirects the browser to the "192.168.0.101/cgi-bin/caos.py" address (2). This path corresponds to the "caos.py" Python script, which is executed by the server (CAOS) and returns an HTML file with several forms that the user can fill. An example of the HTML file rendered at the user's browser is illustrated in Figure B.3. This HTML file is dynamically constructed using a HTML template file (4) and the default parameters of the CAOS, which are stored in a text file named "config.caos" (3).

The configurable parameters depicted in Figure B.3 are:

- # of packets: number of packets to be transmitted;
- Channel: Wi-Fi channel where the packets shall be transmitted;
- Sleep time: Interval in microseconds between packet transmission attempts;
- Length: size of the packet's payload.

The "noisify!" button initiates the transmission of a noise sequence employing the configured parameters. When this button pressed, the Python script invokes the "btx" command line application (5) with the parameters selected in the HTML forms. This file results from compiling the "tx.c" C source code file (B) using the "maketest.sh" script (A). The "tx.c" (B) source file holds the command line application that enables the transmission of IEEE 802.11 packets. This application was developed using the Loss Of Radio CONnectivity (LORCON) library version 2 [168]. The LORCON is an open source library that provides driver abstraction and allows injecting IEEE 802.11 packets into the medium using standard wireless network adapters.

The LORCON library supports the Pedro Larbig's RT73 driver [169], which is compatible with the BELKIN F5D7050 Wi-Fi adapter and exhibits a small access latency, making it an ideal choice for supporting the CAOS tool. In this sense, besides installing the LORCON library, the RT73 driver was also added to the Mint Linux 10 operating system.

B.2 Evaluation

This section aims at studying to which extend is the CAOS tool able to fulfill the requirement of fully occupying the 2.4GHz ISM band. Provided the requirement of flooding the entire 2.4 GHz ISM band with IEEE 802.11 traffic, the CAOS packet transmissions were configured to be performed in multiple IEEE 802.11 channels covering the full range of the 2.4 GHz ISM band according to the settings defined in Table B.1. As documented, packet transmissions are performed with a period of 1 milliseconds in channels 1, 4, 7, 10 and 14, identified as the best compromise between reducing cross-interference and maximizing bandwidth coverage. Since each USB Wi-Fi adapter is factory set with a specific maximum transmission power, the spectral occupation of the 2.4 GHz band will not be uniform across

Table B.1: CAOS settings

Traffic					
Packet size	50 octets = PLCP (24) + MPDU (26)				
Packet period	1 ms				
Data rate	PLCP @ 1 Mbps / MPDU @ 2 Mbps				
Power					
Channel	1	4	7	10	14
Value (dBm)	20	15	16	14	20

the full spectrum. This is observed in Table B.1, where different levels of transmission power are setup for the different network adapters, despite of their common configuration to use a +20 dBm transmission power. The illustrated transmission power values were obtained using the Linux command line "iwconfig" application.

The remaining of this section focuses on assessing the CAOS tool ability to occupy the overall 2.4 GHz ISM band with Wi-Fi noise. As introduced, two approaches are studied. One were a spectrum analyzer is used to verify the band occupation and the other to assess the CAOS electiveness in blocking IEEE 802.15.4 traffic.

B.2.1 Bandwidth occupation on the 2.4 GHz ISM band

The evaluation of the CAOS tool bandwidth occupation in the 2.4 GHz ISM band was conducted using the Aaronia Lcs Analyzer v1.9.9 application together with a SPECTRAN HF2025E v2.2 spectrum analyzer, including the 201 option (Real-Time Broadband Peak Power Meter). The spectrum analyzer was configured with the settings documented in Table B.2. The evaluation was realized for a period of 225 seconds with the CAOS tool performing the transmission of a periodic packet stream in the five channels according to the settings of Table B.1.

The results obtained with the Aaronia Lcs Analyzer application were segmented in two parts. The first corresponds to the (RMS) power measurement during a full sweep of the 2.4 GHz ISM band. In this case, two traces are shown, as depicted in Figure B.4. The darker trace represents last power level sweep, while the lighter registers the maximum power during a trial (225 seconds). Besides the traces, Figure B.4 includes overlay information that helps to read its contents, namely the regions occupied by each Wi-Fi channel and the parameters configured for the spectrum analyzer.

The second part of the CAOS band occupation analysis focuses on the waterfall fre-

Table B.2: Aaronia Lcs analyzer settings

Range (MHz)			
Start	Stop	Span	Center
2400	2490	90	2445
Sweep			
Sampletime		Pulsemode	
1 s		`yes`	
Bandwidth			
Resolution	Video	Attenuator	
3 MHz	3 MHz	`AUTO`	

quency graphic illustrated in Figure B.5. As shown, it encompasses a timestamped rolling window, where each line corresponds to a full sweep of the 2.4 GHz ISM band. In this case, the high power levels are represented in a darker tone. The white gaps indicate that no energy was detected on the sampled frequency.

The maximum power measured in channels 1, 4, 7, 10 and 14 (Figures B.4 and B.5) seems to be aligned with the power supported by the USB Wi-Fi network adapters of Table B.1. However, as shown in Figure B.5, there are visible "holes" in the spectrum, which indicates that the corresponding frequencies had no significant level of energy when the spectrum analyzer performed the sweep. One possible cause for this problem is the occurrence of cross-interference between transmissions. Provided that the selected channels experience some frequency overlapping, the packet transmission rate in "neighbor" channels can be decreased. Furthermore, since the Wi-Fi network adapter's maximum power is hardware dependent, the CAOS tool spectral occupation profile will only be uniform if the selected Wi-Fi USB cards are able to support a similar level of maximum transmission power.

B.2.2 Impact on IEEE 802.15.4 broadcast transmissions

The assessment of the CAOS noise impact on IEEE 802.15.4 broadcast transmissions is an alternative way of testing its effectiveness in mimicking heavily "polluted" Wi-Fi environments. In this sense, a simple testbed encompassing one IEEE 802.15.4 transmitter (T) station (height $\approx$ 78 centimeters), one IEEE 802.15.4 receiver (R) station (height $\approx$ 82 centimeters) and one CAOS (C) tool (height $\approx$ 64 centimeters) was built. The transmitter and receiver were separated by a distance of 3 meters. The CAOS tool was placed at 3.9 meters from the central point between transmitter and receiver. Figure B.6(a) shows an

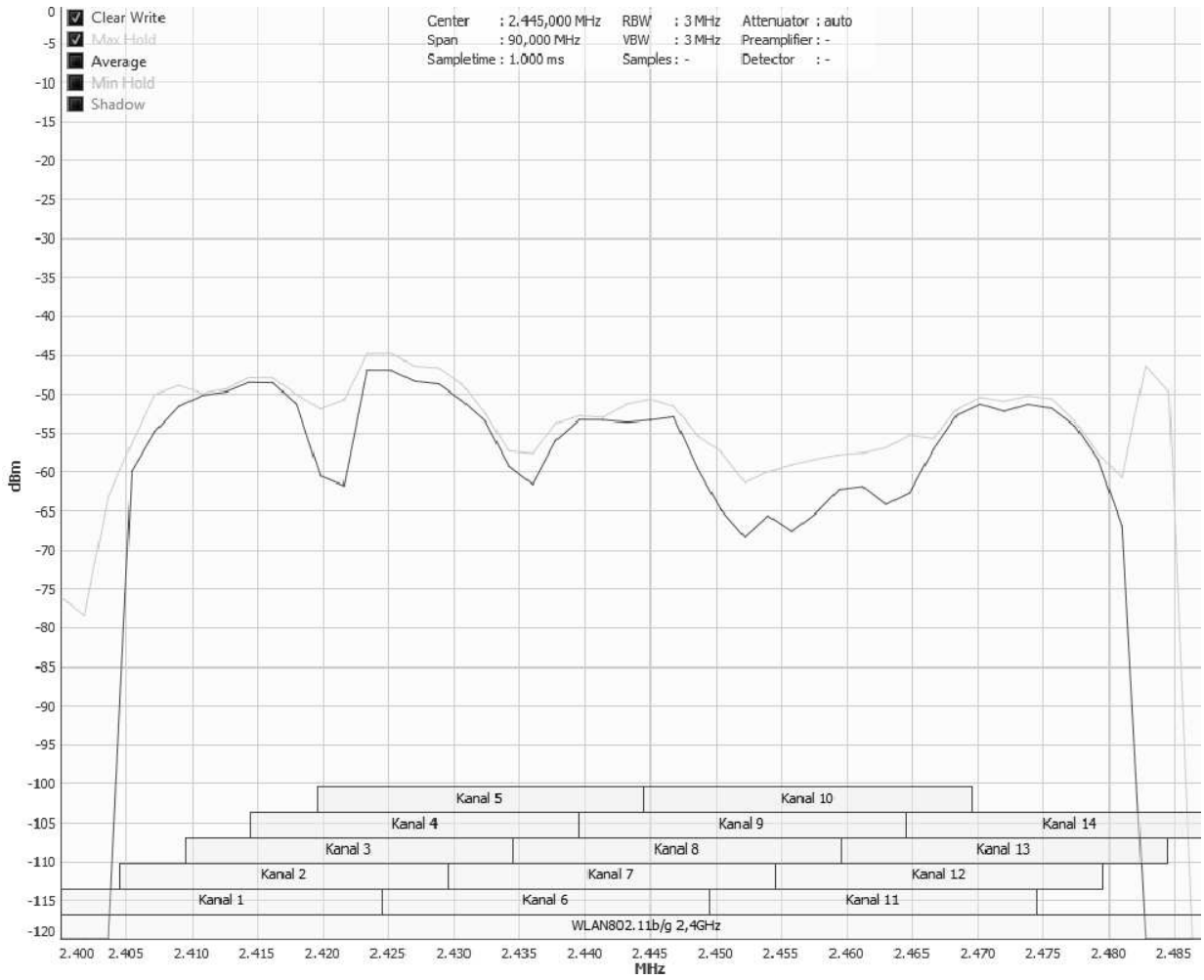


Figure B.4: CAOS frequency sweep

illustrative diagram of the testbed's physical arrangement and Figure B.6(b) provides a photography of its realization.

The CAOS assessment was based on trials, each one comprehending the transmission of 1000 IEEE 802.15.4 broadcast packets with a payload of one byte. Both transmitter and receiver stations were programmed with a firmware application allowing their configuration/control through a terminal emulator. Using this approach, the IEEE 802.15.4 transmitter station was configured to employ a power of -10 dBm and to conduct transmission attempts at every 100 milliseconds. Each transmitted packet carried one distinct character symbol that was also sent to the attached terminal emulator. Since the sequence of characters was known *a priori*, the receiver station could determine when a packet was lost. The successfully received symbols and the '_' character, representing the failed re-

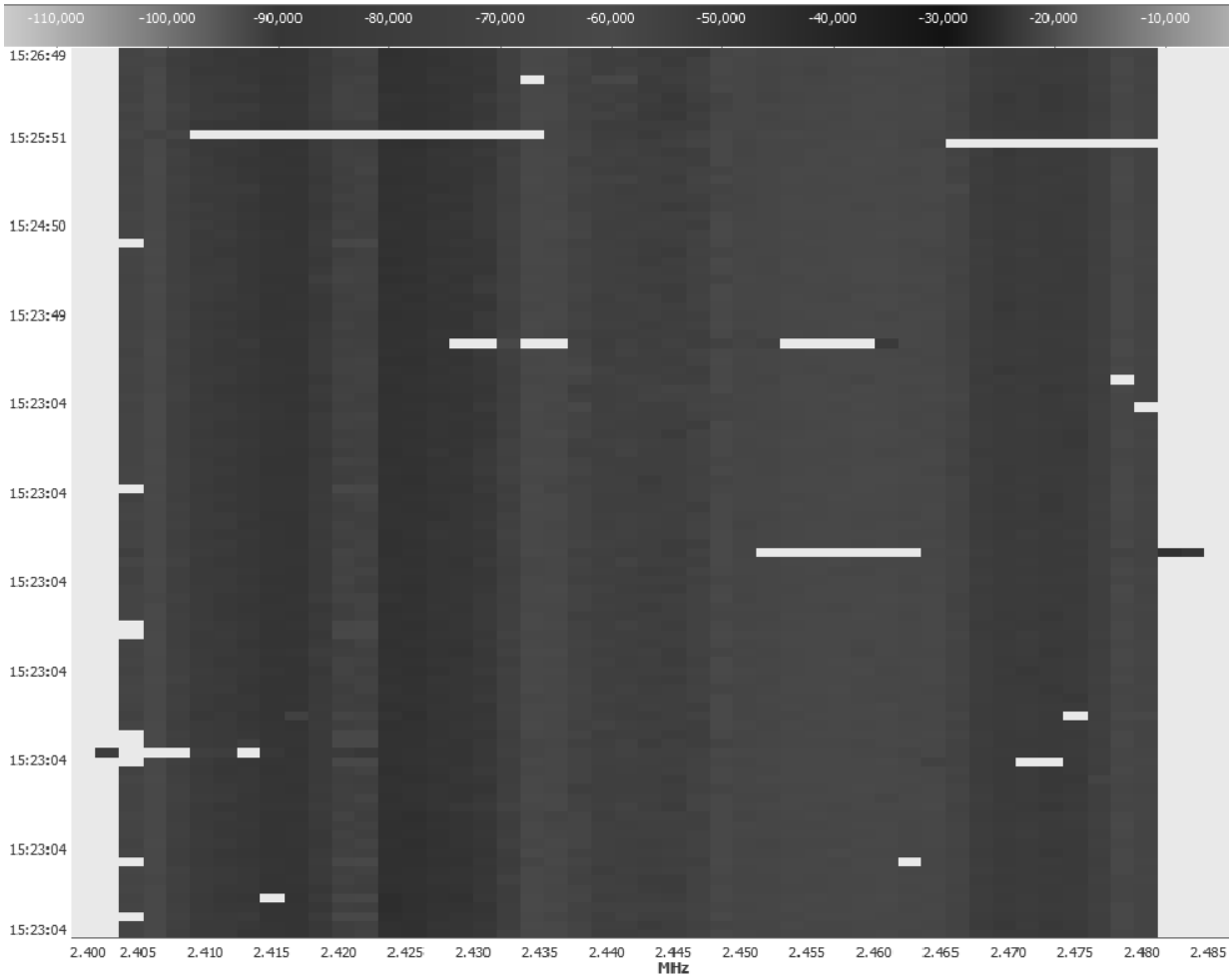
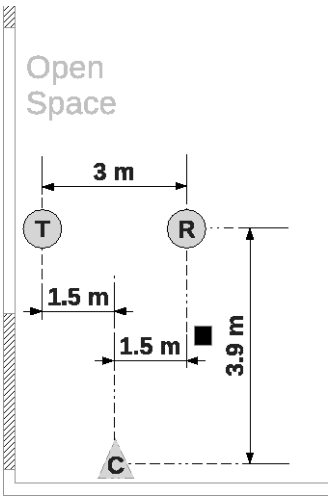


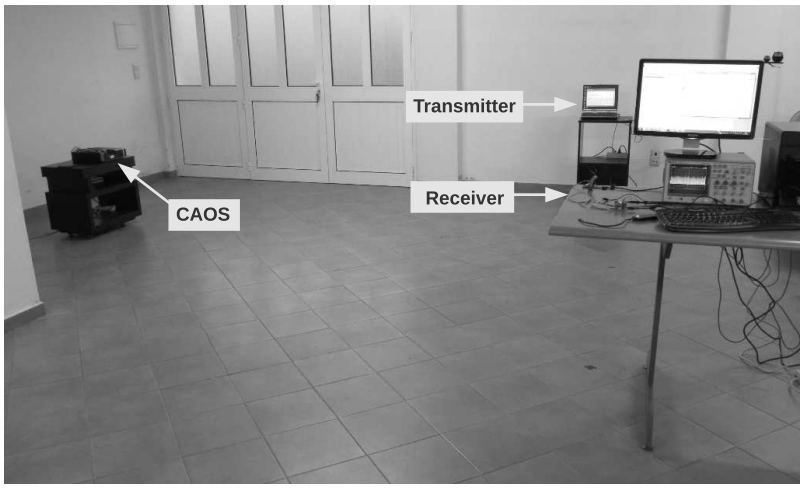
Figure B.5: CAOS frequency occupation waterfall

ceptions, were printed to the terminal emulator. By counting the number of existing '_' characters at the receiver, it was possible to determine the transmission's packet error rate (PER).

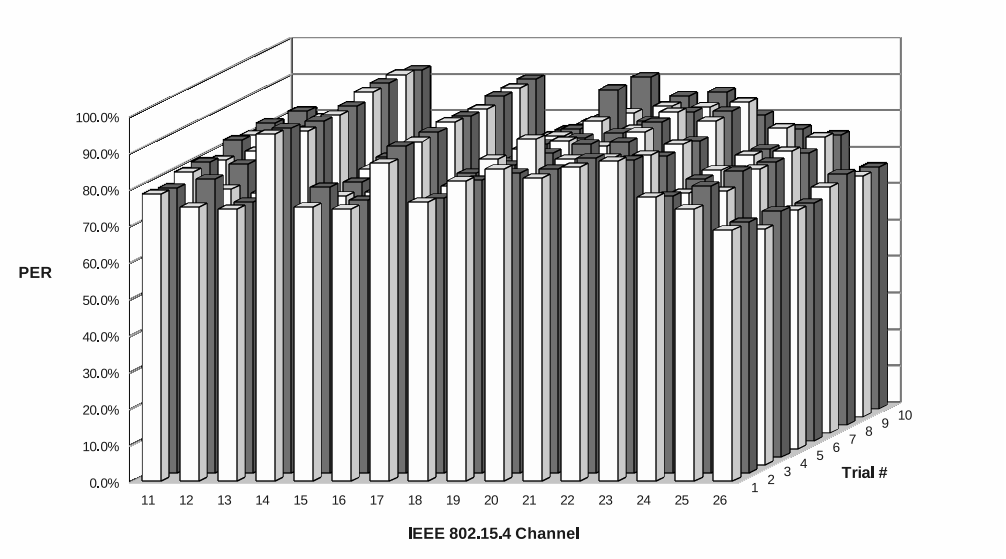
In order to measure the impact of the CAOS noise over the full length of the 2.4 GHz ISM band, 10 trials were conducted for each one of the 16 IEEE 802.15.4 channels. As documented in Figure B.6(c), the obtained results were combined to create a 3D graphical representation, illustrating the experienced IEEE 802.15.4 PER as a function of the channel and of the trial number. These results show that the noise produced by the CAOS tool forces a high PER (typically more than 70 %) on the IEEE 802.15.4 broadcast transmissions, independently of the channel being used to perform them. Although the PER is not uniform across all IEEE 802.15.4 channels, a consistent behavior between different



(a) Physical arrangement



(b) Setup photo



(c) Results

Figure B.6: CAOS noise impact evaluation setup and results

trials can be observed. This effect is possibly caused by the different levels of transmission power employed by the Wi-Fi network adapters embedded in the CAOS tool.

C

The uMRF Wireless Platform

This appendix describes the uMRF wireless platform whose elements were employed in the development of the testbeds created to assess the *bandjacking* technique and the WFTT protocol performance. The platform explanation is divided in two sections, one dedicated to the wireless device used in the implementation of the WFTT slave (uMRFs) and the other addressing the device employed by the master station (uMRF).

C.1 uMRFs: A Tiny Wireless Node

This section presents the uMRFs wireless device fact sheet, whose development specifically targets the support of low-power wireless networks based on the IEEE 802.15.4 standard. The design of this device was driven by the functionality required for implementing a WFTT slave station.

C.1.1 Overview

The uMRFs board is an integrated solution for developing wireless applications based on the IEEE 802.15.4 standard. Besides the obvious communication functionality, the uMRFs board encompasses several hardware components, allowing extending its usage to monitor temperature and acceleration, for example. Given its low energy consumption and small size, the uMRFs board can enable mobile applications when used together with a Li-IoN battery. The circuitry for charging the battery from a USB power source is also included in the board.

The uMRFs board encompasses several components packed in a tiny board to provide the following set of features:

- Micro-controller operating at up to 16MIPS (Microchip PIC18 family);
- IEEE 802.15.4 standard low-power wireless radio module;
- Low-power digital accelerometer;
- USB or battery powered, with on-board battery charger;
- USB serial port;
- General buttons and LEDs for debug;
- Low power consumption;
- Compact and small profile: 44 mm x 44 mm x 9 mm;
- Broad range of development tools;
- Free software tool-chain for academic use.

C.1.2 Hardware

Figure C.1 shows an enlarged picture of the uMRFs board decorated with information identifying the included components. Tables C.1, C.2 and C.3 document the main characteristics of the microcontroller, wireless module and peripherals, respectively. As documented, the uMRFs board encompasses a 8-bit nano Watt XLP Microchip PIC18F26K20 microcontroller, capable of operating with a frequency of up to 64 MHz (16 MIPS). Besides its low-power operation and flexible frequency setup, the PIC18F26K20 microcontroller boasts 64 kilobytes of program memory, 3936 bytes of data memory and 1024 bytes of EEPROM memory. As documented in Table C.1, the microcontroller supports a wide range communication peripherals, timers and input/output pins for both analog and digital operation.

The Microchip's MRF24J40MA wireless module (Table C.2) integrated on the uMRFs board is responsible for its enabling with IEEE 802.15.4 wireless communications. Besides the IEEE 802.15.4 standard conformance, the module supports the low-power operation sleep mode; the development of applications with ZigBee, MiWi (+P2P) and proprietary communication stacks; and features several distinctive functionalities such as a security engine (AES-128) supporting the CTR, CCM and CBC-MAC modes.

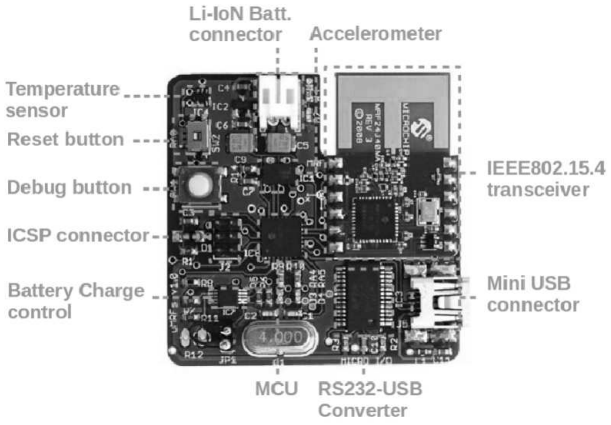


Figure C.1: The uMRFs development board

Table C.1: PIC18F26K20 characteristics

Microcontroller	
Type	8-bit nano Watt XLP
Frequency	Up to 64 MHz (internal oscillator)
Operations per second	Up to 16 MIPS
Program memory size	64 kB
Data memory size	3936 B
EEPROM memory size	1024 B
Analog-To-Digital converter	11 channels of 10-bit
I/O pins	25 I/O + 1 input
Communications	EUSART, SPI, I2C
Timers	1 x 8-bit, 3 x 16-bit
Others	1 x CCP, 1 x ECCP (PWM)

Table C.2: MRF24J40MA characteristics

Wireless Module	
Type	IEEE Std. 802.15.4
Frequency	2.4 GHz
Antenna	On-board, +0 dBm
Data rate	250 kbps
Range	Up to 400 m outdoor / 10 m indoor
Power consumption	Reception: 19 mA, Transmission: 23 mA, Sleep: 2 nA
Supported protocols	ZigBee, MiWi (+P2P), proprietary
Features	Hardware CSMA-CA mechanism; auto ACK; hardware security engine (AES-128) with CTR, CCM and CBC-MAC modes; support all CCA modes and RSS/LQI

Table C.3: uMRFs peripherals' characteristics

Peripherals	
Accelerometer	Freescale MMA7455L (SPI)
Precision	8/10-bit x 3 axis
Scale	+/- 2/4/8g
Power consumption	400 μA
Temperature sensor	Microchip MCP9700
Type	Analog output
Accuracy	4 °C
Battery charger	Microchip MCP73833
USB to serial converter	FTDI FT232RL
Programming interface	5 or 6-wire ICSP
Hardware debug LEDs	2 (orange + green)
Hardware debug buttons	2 (general purpose + reset)

As shown in Table C.3, the uMRFs board offers support for mounting several peripherals, namely: accelerometer, battery charger controller, USB to serial converter, programming interface and several hardware input/outputs. The accelerometer was included to enable shock/fall detection applications, for example. The serial converter allows the uMRFs boards to dialogue with PC applications using serial communications. The support for mobile applications was included by powering the board with a battery and enabling its recharge through a standard USB connection. Finally, the board's hardware inputs and outputs provide a simple mechanism to externally control and monitor its operation.

C.2 uMRF: An Extensible Wireless Board

The uMRF board is an embedded solution for developing IEEE 802.15.4 wireless communication applications. The key design aspect of this board is its flexibility to bridge a low-power IEEE 802.15.4 network to a higher level communication tier implemented in compliance to the IEEE 802.11 (Wi-Fi) or IEEE 802.3 (Ethernet) standards. The original design of this device was driven by the functionality required for implementing a WFTT master station. This section describes the main characteristics and features of the uMRF board.

C.2.1 Overview

The uMRF (read as "micro MRF") board is an integrated solution for developing IEEE 802.15.4 wireless applications. Besides the IEEE 802.15.4 communication function-

ality, which is enabled with either the MRF24J40MA (400 meters range outdoor / 10 meters range indoor) or MRF24J40MB (1200 meters range outdoor / 10 meters range indoor) modules, the uMRF board encompasses two connectors, which allow extending its functionality using mezzanine boards (add-ons) with support for Wi-Fi and Ethernet communications. The small size (less than a credit card), moderate energy consumption and high processing power make the uMRF board a good solution for resource demanding applications where multiple communication technologies are required. A Li-IoN battery can be used to power up the board, which includes the circuitry for charging the battery from a USB power source.

The uMRF embeds several components packed in a credit card sized board to provide the following set of features:

- High performance microcontroller operating at up to 40MIPS (Microchip dsPIC family);
- On-board IEEE 802.15.4 standard low/moderate power radio module;
- USB or battery powered, with on-board battery charger;
- USB serial port;
- General purpose buttons and LEDs for debugging;
- Low power consumption;
- Small size: 69 mm x 49 mm x 9 mm.

C.2.2 Hardware

Figure C.2 shows an enlarged picture of the uMRF board decorated with information identifying the included components. As documented, the board provides soldering points for mounting two IEEE 802.15.4 modules of different sizes. Furthermore, the mounting of some external peripherals such as the temperature sensor and the serial-USB converter is optional, allowing to reduce the board's overall cost.

Tables C.4, C.5 and C.6 document the main characteristics of the microcontroller, wireless module and peripherals, respectively. The uMRF board integrates a 16-bit Microchip dsPIC33FJ256MC710 microcontroller capable of operating with frequencies of up to 80

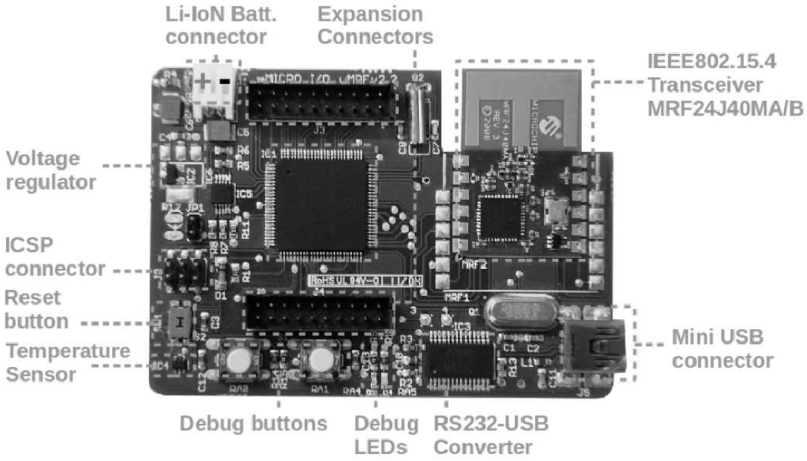


Figure C.2: The uMRF development board

MHz (40 MIPS). In addition to the broad frequency support, this microcontroller comprises a Program memory of 256 kilobytes, a data memory of 30 kilobytes and support for Direct Memory Access (DMA). Table C.4 shows that, when compared to the PIC18F26K20 microcontroller adopted for the uMRFs board, the dsPIC33FJ256MC710 MCU is enabled with a much wider range of communication peripherals, timers and input/output pins for both analog and digital operation.

Table C.4: dsPIC33FJ256MC710 characteristics

Microcontroller	
Type	16-bit
Frequency	Up to 80 MHz
Operations per second	Up to 40 MIPS
Program memory size	256 kB
Data memory size	30 kB
Memory features	Direct Memory Access (DMA)
Analog-To-Digital converter	24 channels x 2
I/O pins	85
Communications	2 x UART, 2 x SPI, 2 x I2C, 2 x CAN
Timers	9 x 16-bit, with pairing: 4 x 32-bit
Others	8 x PWM

As shown in Table C.5 the IEEE 802.15.4 MRF24J40MB module presents similar characteristics to the MRF24J40MA module, except for the higher range and increased current consumption. The range extension is achieved through the inclusion of both a power amplifier, increasing the propagated energy level, and and low noise amplifier, augmenting the reception sensitivity.

Table C.5: MRF24J40MB characteristics

Wireless Module	
Type	IEEE Std. 802.15.4
Frequency	2.4 GHz
Antenna	On-board, +20 dBm
Data rate	250 kbps
Range	Up to 1200 m outdoor / 100 m indoor
Power consumption	Reception: 25 mA, Transmission: 130 mA, Sleep: 5 μA
Supported protocols	ZigBee, MiWi (+P2P), proprietary
Features	Hardware CSMA-CA mechanism; auto ACK; hardware security engine (AES-128) with CTR, CCM and CBC-MAC modes; support all CCA modes and RSS/LQI

The uMRF board includes peripherals such as a temperature sensor, a USB to serial converter, a battery charger controller and some hardware inputs and outputs. However, its distinctive functionality arises from the possibility of mounting mezzanine boards holding other types of communication interfaces. The support of the IEEE 802.11 (Wi-Fi) and the IEEE 802.3 (Ethernet) standard communication technologies was achieved through the development of specific mezzanine boards. Another example of the functionality extension conveyed by the use of mezzanine boards is the PNS connection to the WFTT master station. In this case, a mezzanine board is used to bridge the driving signals of the master to the PNS board.

Table C.6: uMRF peripherals' characteristics

Peripherals	
Temperature sensor	Microchip MCP9700
Type	Analog output
Accuracy	4 °C
Battery charger	Microchip MCP73833
USB to serial converter	FTDI FT232RL
Programming interface	5 or 6-wire ICSP
Hardware debug LEDs	2 (orange + green)
Hardware debug buttons	3 (2 x general purpose + reset)

D

BeeMon: A IEEE 802.15.4 2.4 GHz energy monitor

The motivation for developing an IEEE 802.15.4 channel energy monitor is the requirement of evaluating the periods of time in which IEEE 802.15.4 transmissions occupy the medium. This can be an important tool to assess the timeliness of any wireless protocol using the IEEE 802.15.4 technology to perform transmissions. This appendix describes the hardware of the BeeMon, an IEEE 802.15.4 2.4 GHz channel energy monitor, with emphasis on its hardware and software. Its operation is also described in detail.

D.1 Architecture

The architecture of the BeeMon monitor builds on the features provided by the uMRFs board presented in Appendix C.1. The availability of a platform including a suitable MCU and an IEEE 802.15.4 transceiver that is capable of performing fast energy measurements was essential in the development of a low cost monitoring solution for the 2.4 GHz ISM band.

D.1.1 Hardware

As depicted in Figure D.1, the BeeMon monitor operates with the help of an oscilloscope, which allows visualizing a signal proportional to the energy levels detected in the medium. In this sense, the BeeMon monitor produces an analog signal matching the energy sampled on a given channel or on a set of channels, as it will be explained further ahead. The hardware of the BeeMon monitor encompasses a uMRFs board and a low pass filter.

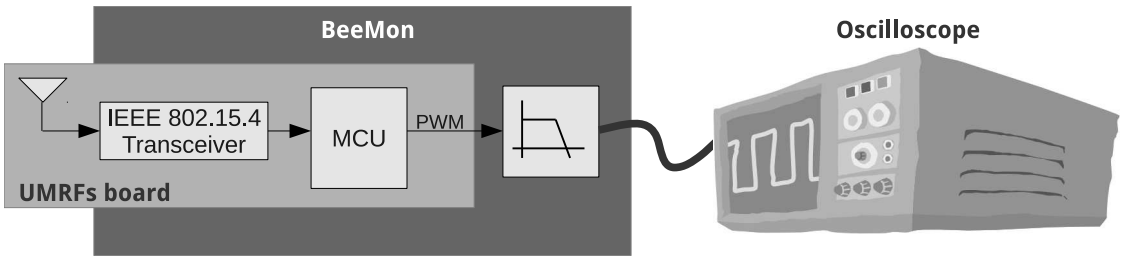


Figure D.1: Architecture of the BeeMon IEEE 802.15.4 channel monitor

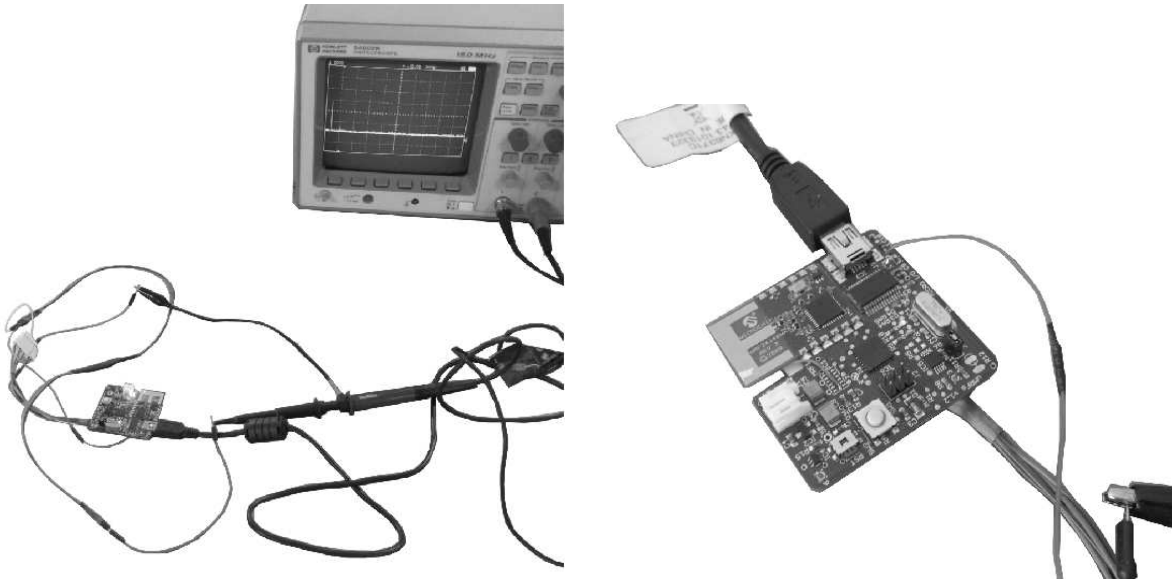
As detailed in Appendix C.1, the uMRFs board uses a Microchip PIC18F26K20 to run the application firmware that, among other functions, drives the board's MRF24J40MA transceiver. Provided that this MCU does not have an integrated DAC, a PWM output was used to generate an energy proportional digital signal, which can be converted into an analog signal that can be visualized in an oscilloscope.

The low-pass filter allows converting the PWM signal from the MCU into a proportional analog counterpart. The filter is a simple RC circuit designed with a cutoff frequency of 3183 Hz and a time constant of 50 microseconds. Therefore, the response to a step input signal takes around 5.3 microseconds to reach 10 % of its amplitude, which is the minimum perceivable change in the output signal using an oscilloscope. Hence, a perceivable change on the BeeMon output analog signal will be affected by a constant delay of approximately 5.3 microseconds introduced by the low-pass filter. This delay must be accounted on the overall BeeMon monitor response time.

The physical implementation of the BeeMon monitor is shown in Figure D.2. The uMRFs board is powered by a USB connection that, additionally, allows sending serial commands to change the mode of operation or channel of the BeeMon monitor. The RC low-pass filter was constructed and embedded in the wire that is connected to the MCU's PWM output. Figure D.2(b) shows a separate wire that carries the PWM signal to the low-pass filter.

D.1.2 Software

The design of the BeeMon monitor relies heavily on the features provided by the MRF24J40 transceiver to sample the energy levels of a IEEE 802.15.4 channel. All interactions with the MRF24J40 transceiver via the SPI bus were optimized in the firmware by reducing to a minimum the number of function callings, i.e., by sending commands to the MRF24J40 transceiver in the most low level possible way, thus avoiding the latency of



(a) Overview (b) Detail

Figure D.2: Physical implementation of the BeeMon monitor

context save and restore operations associated to the cascade invocation of functions.

The key operations of the firmware developed for the BeeMon monitor are the energy sampling using the MRF24J40 transceiver [153] and the conversion of the samples into an analog signal that can be visualized in an oscilloscope. In this sense, the first step is then the collection of digital values proportional to the energy in the medium. This task was realized using the energy detection feature of the MRF24J40 transceiver. According to its datasheet, the period of time used to sample the energy in any channel can be configured to last for 1, 2 , 4 or 8 symbols, the later being the default value. This means that the energy detection algorithm will provide an energy value in the MRF24J40 RSSI register containing the averaged RSSI received power levels over this period of time. In this BeeMon implementation, a two symbol configuration was chosen, making the channel sample period last for 32 microseconds.

The second key operation of BeeMon's firmware is the digital to analog conversion of the sampled levels of energy. As introduced, the MCU's PWM module is used together with a low-pass filter for this purpose. Hence, the PWM register that controls the signal's duty-cycle is loaded with the 8-bit value (0 to 255) of sampled energy stored in the MRF24J40's RSSI register. The PWM signal was configured with a 16.125 microseconds period. An important aspect regarding the PWM generation timeliness is the fact that any value loaded

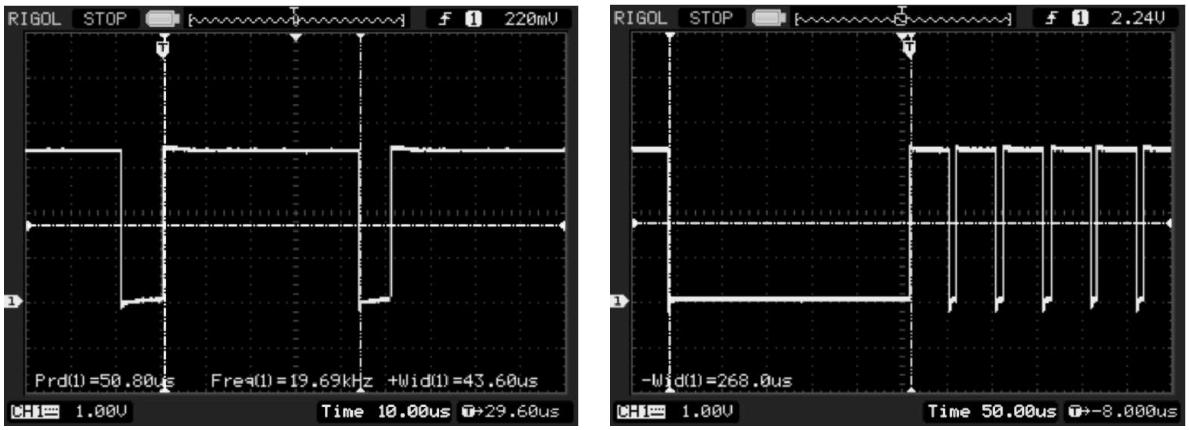
into the duty cycle register is only latched after the current PWM period completes. This means that there can be a delay of a maximum of two PWM periods between the loading of the sampled energy level into the duty cycle register and its effectiveness (one complete PWM period) in the output PWM signal. As mentioned above, after being generated, the PWM signal is fed to a low-pass filter that outputs a proportional analog signal to visualize in a oscilloscope.

The latency associated to the process of sampling the medium energy and converting it to an analog signal whose variations are visible on an oscilloscope is not negligible. This latency corresponds to the sum of the following delays:

1. Sample time at the MRF24J40 transceiver;
2. Communication time of the averaged RSSI power from the transceiver to the MCU;
3. Update time of the PWM duty cycle register;
4. Latch time of the PWM duty cycle value;
5. PWM period with the updated duty cycle value;
6. Low-pass filter time to produce a visible a variation on the output analog signal.

In order to have a more experimental estimate of the BeeMon monitor response time, an evaluation of its firmware implementation was performed. In this evaluation, a digital output pin is set between the instants where the energy sampling is triggered and the instant where the corresponding value is updated in the PWM duty cycle register. The digital pin signal is tied to a Rigol DS1052E oscilloscope for its characterization. As shown in Figure D.3(a), the experimental sum of the delays 1, 2 and 3 is, approximately, 43.6 microseconds. Provided that the delays 4,5 and 6 have been already presented, the overall response time of the BeeMon monitor is 64.725 microseconds with a maximum jitter of 16.125 microseconds, due to the phase difference of the duty cycle register latching. In order to simplify the analysis of the timings collected with the BeeMon, its response time is assumed to encompass a delay of 65 microseconds and a jitter of 16 microseconds.

Due to the fact that some of the referred operations can be parallelized, e.g., the energy sampling on the MRF24J40 transceiver can be simultaneously conducted with the latching of a PWM duty cycle value of a former energy sample, the sampling period of the BeeMon monitor can be highly reduced. In this sense, in order to have a fixed sampling period and provide enough time to the MCU to perform other tasks, a 50 microsecond sampling



(a) Sample period

(b) Channel change slack

Figure D.3: BeeMon timing constraints

interval was defined, thus allowing an energy sampling frequency of 20 KHz. The sampling period can be seen in Figure D.3(a).

The BeeMon supports two types of monitoring operations: *single channel* and *channel sweep*. The first was already discussed and corresponds to the continuous energy monitoring on an IEEE 802.15.4 channel. The second performs a cyclical energy monitoring in all IEEE 802.15.4 channels of the 2.4 GHz ISM band. One of the main limitations of implementing such a feature is the requirement of guaranteeing that the internal oscillator is stable before initiating any interactions with the transceiver. According to the datasheet of the MRF24J40 transceiver [153], a wait period of 192 microseconds should be enforced after changing channel. A delay of approximately 268 microseconds is used in this implementation, as it can be seen in Figure D.3(b). The sweep speed of the BeeMon monitor is set in the firmware by means of *define* instructions. Currently, provided that the sampling period of a channel is 50 microseconds and that there is a significant delay for exchanging the sampling channel, eight samples per channel are collected. The maximum value of each set of samples is then stored into a data array with 16 positions, one for each channel. Then after each complete channel sweep, the values stored in the data array are fed to the PWM module, which then, together with the low-pass filter generates an analog signal representing the energy levels observed in the IEEE 802.15.4 channels. The switching between the two modes of operation is performed by serial commands sent to the BeeMon monitor according to the predefined format presented in the following section.

(a) Energy over time on a given channel (b) Energy over the 16 IEEE 802.15.4 channels

Figure D.4: BeeMon visualization modes

D.2 Operation

The visualization of the signals representing the energy levels on the medium requires some adjustments in the oscilloscope. In this sense, in the single channel mode of operation, the oscilloscope should be set to the Y-T mode (or Roll mode in case it is available) with a scale of between 200 mVdiv e 1 Vdiv. Figure D.4(a) shows the channel energy analysis using the BeeMon monitor of a IEEE 802.15.4 communication protocol. As illustrated, three transmissions are visible over a period of 60 milliseconds.

In the channel sweep mode, all IEEE 802.15.4 channels are sampled sequentially. In order to synchronize the oscilloscope with the BeeMon channel sweep, the external trigger of the oscilloscope should be connected to the trigger signal of the BeeMon monitor. This trigger signal generates an impulse (3.3 V amplitude) that marks the beginning of the channel sweep operation. Afterwards, the output of the BeeMon monitor represents the energy sampled in the IEEE channels, starting in channel 11 and ending in channel 26. All channel sweeps are preceded by this trigger impulse to allow the synchronization of the attached oscilloscope. The duration of each energy slot corresponding to a given channel is similar to the duration of the synchronization impulse. Figure D.4(b) depicts the trigger signal in blue and the energy per channel in yellow. As shown, energy was detected in channels 22, 23, 25 and 26, the latter with the highest level. Using the oscilloscope's display persistence feature, the energy levels of the previous channel sweeps can be visualized.

The BeeMon monitor can be configured using simple commands, which are sent via a

serial connection configured with a baud-rate of 115200 bps, eight data bits, no parity and one stop bit (8N1). The supported commands are listed in Table D.1. As shown, the user can initiate or pause the BeeMon's operation by sending the characters 'i' or 'p' through the serial connection, respectively. Furthermore, the BeeMon monitor supports commands to reset its operation, switch between (single channel/multi channel) operation modes and set the operation channel (only in the single channel mode).

Table D.1: BeeMon serial commands

Command	Description
i	(i)nitiate operation
p	(p)ause operation
t	(t)oggle (start/stop) operation
r	(r)eset BeeMon hardware
s	(s)ingle channel mode
m	(m)ultiple channel sweep mode
0-f	Set operating channel from 11 to 26 (single channel mode)

Although the BeeMon monitor has known timeliness limitations, it can be a helpful tool to analyze and debug time sensitive communication protocols that make use of the IEEE 802.15.4 technology.