UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática

INACCESSIBILITY IN WIRELESS SENSOR NETWORKS

André Alexandre Margarido Taborda Vaz Guerreiro

DISSERTAÇÃO

MESTRADO EM ENGENHARIA INFORMÁTICA
Especialização em Arquitectura, Sistemas e Redes de Computadores

2013

UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática

INACCESSIBILITY IN WIRELESS SENSOR NETWORKS

André Alexandre Margarido Taborda Vaz Guerreiro

DISSERTAÇÃO

MESTRADO EM ENGENHARIA INFORMÁTICA
Especialização em Arquitectura, Sistemas e Redes de Computadores

Dissertação orientada pelo Prof. Doutor José Manuel de Sousa de Matos Rufino
e co-orientada pelo Mestre Jeferson Luiz Rodrigues Souza

2013

Acknowledgments

During the course of the work which led to the present dissertation, including the curricular part of the Masters programme, there are many people to which I am most thankful for all the different kinds of support they gave me. May my gratitude towards them remain forever patent in these lines.

Firstly, I would like to thank my advisor, Prof. José Rufino and my co-advisor Jeferson L. R. Souza, for they crucial help during the development of this thesis. Their remarks and suggestions have always contributed to the increase of my knowledge and to develop further my work. Theirs dedication and support while I was writing this thesis and related articles was truly important. I think i couldn't get better orientation than which was given to me. Their availability, good advices, and friendship were extremely important for me.

Secondly I would like to thank Navigators research group and all of the LaSIGE fellows for the interesting and constructive discussions where I learnt a lot.

Thirdly, I would like to thank all my colleagues and friends from Lab 8.2.25, for their help and support, and priceless moments of fellowship.

Last, but not least, the biggest "Thank you"of the world goes for my family. My mother Elsa, my father Manuel and my grandmother Maria. They have always been there for me and with their motivation and enthusiasm forced me to keep working even in the hardest moments, and to always choose my own way to achieve the proposed goals. It is not too much to say that I couldn't do this without them.

This work was partially supported: by the EC, through project IST-FP7-STREP-288195 (KARYON); by FCT/DAAD, through the transnational cooperation project PROPHECY; and by FCT, through the project PTDC/EEI-SCR/3200/2012 (READAPT) and the Multiannual Funding Program.

À minha família e amigos.

Abstract

Wireless networks are seen as the communication networks of the future, providing communication capabilities where cables are not able to be used. Wireless technologies enable network flexibility and mobility, and reduce size, weight, and power consumption (SWaP) of communication devices. The IEEE 802.15.4 standard was designed to support the specification of wireless sensor networks (WSNs) and wireless sensor and actuator networks (WSANs), where is emerging their utilization within environments with real-time requirements, such as industrial and aerospace.

The medium access control (MAC) layer is the control foundation of the network communication services. Disturbances in the MAC layer operation may lead to a network inaccessibility scenario, which consists in a temporary absence of network communication although the network is not considered failed. Examples of such disturbances are electromagnetic noise interference, glitches in the wireless device circuitry, or even obstacles in the communication path.

A previous theoretical study indicates the occurrence of periods of network inaccessibility as a source of MAC transmission protocol delays which may induce application deadline misses which that compromise the dependability and timeliness properties of the whole networked system. Thus, this work aims to validate that previous study using the network simulator NS-2.

The NS-2 simulator is a widely used tool supporting the simulation of IEEE 802.15.4 wireless networks. However, we discovered that its compliance to the IEEE 802.15.4 standard is imperfect. In order to perform the validation of the theoretical characterisation of network inaccessibility new mechanisms need to be introduced in the IEEE 802.15.4 simulation model. These improvements comprises: the support for real-time transmissions, through the incorporation of the contention free period (CFP) and of guaranteed time slot (GTS) ; IEEE 802.15.4 standard management operations not implemented in the official NS-2 release; A flexible tool capable of re-create the inaccessibility events and simulate different error conditions on the network, which include the Fault Injector and temporal and energetic analysis tool;

Keywords: wireless networks; network inaccessibility; dependability; timeliness; real-time;

Resumo

As redes sem fios têm sido encaradas como as redes de comunicação do futuro, fornecendo capacidades de comunicação onde os cabos não podem de ser utilizados. As tecnologias sem fio permitem flexibilidade e mobilidade na rede como também reduzir o tamanho, peso e consumo energético (SWaP) dos dispositivos de comunicação.

A norma IEEE 802.15.4 foi projetada para suportar a especificação de redes de sensores sem fio (WSNs) e redes de sensores e atuadores sem fios (WSANs), e a sua utilização está a emergir em ambientes com requisitos de tempo real, tais como o industrial e aeroespacial.

A camada de controlo de acesso ao meio (MAC) é o alicerce de controlo dos serviços de comunicação da rede. Distúrbios no funcionamento desta camada podem levar a rede a entrar num estado apelidado de inacessibilidade, este caracteriza-se numa falta temporária de comunicação na rede, embora não se considere que a rede falhou. Exemplos de tais perturbações são ondas eletromagnéticas, falhas no circuito de dispositivos sem fios, ou até mesmo obstáculos no caminho de comunicação.

Um estudo teórico anterior indica a ocorrência de inacessibilidade como fontes de atraso portanto, falhas no cumprimento de prazos que podem comprometer propriedades de confiabilidade e pontualidade de todo o sistema. Assim, este trabalho tem como objetivo validar que o estudo anterior, utilizando o simulador de rede NS-2.

O simulador de rede NS-2 é uma ferramenta amplamente utilizada no suporte a simulação de redes sem fio IEEE 802.15.4. No entanto, descobrimos que não se encontra totalmente em conformidade com a norma IEEE 802.15.4. Com o intuito de efetuar a validação dos modelos de inacessibilidade, novos mecanismos devem ser introduzidos no modelo de simulação referente ao IEEE 802.15.4. Estes melhoramentos compreendem: Suporte para transmissões de tempo real, através da incorporação do mecanismo de acesso livre de contenção (CFP) e do intervalo de tempo de acesso garantido (GTS); Desenvolver as operações de gestão normalizadas não concretizadas no modulo IEEE 802.15.4 presente na versão oficial do NS-2;Adição de novos recursos necessários para a avaliação da rede em condições de erro, mais especificamente, um injetor de faltas, e um módulo de contabilização temporal e energético.

Palavras-chave: redes sem fios; inacessibilidade; confiabilidade; pontualidade; tempo-real;

Resumo alargardo

As redes sem fios têm sido encaradas como as redes de comunicação do futuro, fornecendo capacidades de comunicação onde os cabos não podem de ser utilizados. As tecnologias sem fio permitem flexibilidade e mobilidade na rede como também reduzir o tamanho, peso e consumo energético (SWaP) dos dispositivos de comunicação. Devido às suas características únicas, há um grande interesse no desenvolvimento de aplicações que utilizem Redes de Sensores Sem Fios e Redes de Sensores e Actuadores Sem Fios em diferentes sectores, tais como monitorização de recursos naturais, aeroespacial, automóvel e industrial. A maioria destes ambientes têm restrições de comunicação em tempo real, o que implica que as Rede de Sensores Sem Fios e as Redes de Sensores e Actuadores Sem Fios devem ser capazes de fornecer suporte a serviços de comunicação em tempo real e dar garantias acerca dos limites do tempo de transmissão.

No entanto, o meio de comunicação aberto e partilhado das redes sem fio é altamente suscetível a interferências eletromagnéticas, e a obstáculos existentes no caminho da comunicação. Estes problemas podem perturbar as comunicações realizadas pela camada de controle de acesso ao meio (MAC). Melhorar a previsibilidade temporal e a confiabilidade dos serviços de nível MAC é de extrema importância de forma a proporcionar um serviço de transmissão de dados em tempo real eficiente através das redes sem fios.

Existem diversos estudos focados no suporte de serviços de comunicação fiável e tempo real em comunicações em redes sem fios, mais propriamente ao nível mais baixo da pilha de protocolos de comunicação. Contudo esses estudos dão pouca ou nenhuma importância aos aspetos de fiabilidade da camada de acesso ao meio e seus serviços. No entanto a confiabilidade e a pontualidade são essenciais para assegurar a capacidade de resposta e recuperação da normal operação da rede quando esta é sujeita a condições de erro.

Tais erros podem afetar a operação da camada MAC e induzir paragens temporárias da rede, um fenómeno que designamos por inacessibilidade, e que pode impedir a operação da rede em tempo real.

Um estudo teórico anterior indica a ocorrência de inacessibilidade como fontes de atraso portanto, falhas no cumprimento de prazos que podem comprometer propriedades de confiabilidade e pontualidade de todo o sistema.

Este estudo representa uma motivação para a investigação em curso que aborda o teste e avaliação de redes de sensores sem fios e redes de sensores e actuadores sem fios através do uso de simuladores de rede, nos quais adoptamos a norma IEEE 802.15.4 e o seu conhecido potencial para suportar o tráfego de tempo real (através da atribuição de acesso à rede exclusivo), como um caso de estudo.

A utilização de simuladores representa uma forma adequada de testar e avaliar a dinâmica de uma rede em diferentes condições ambientais. Existem vários simuladores de rede disponíveis, alguns com licença comercial, como o OPNET, e outros com código-fonte aberto ou de licença académica, como Omnet++ , Prowler, TOSSIM e NS-2. O simulador de rede NS-2 é das ferramentas de simulação mais amplamente aceites e utilizadas na comunidade científica, é uma plataforma de código-fonte aberto e de arquitetura modular, suporta a simulação de redes de sensores sem fios e redes de sensores e actuadores sem fios através da norma IEEE 802.15.4.

Embora o NS-2 possua um módulo respeitante a norma IEEE 802.15.4, este módulo não vem com um suporte nativo para aplicações com necessidades de comunicação em tempo real, como por exemplo a simulação de um período livre de contenção (CFP) no qual é possível alocar intervalos de tempo para acesso exclusivo a rede.

É o nosso objetivo ultrapassar a limitação existente no módulo IEEE 802.15.4 do simulador NS-2, que, originalmente, apenas permite comunicações baseadas em contenção.

Assim, este trabalho apresenta melhorias no módulo IEEE 802.15.4 NS-2 de forma a que este proporcione melhor suporte ao teste, simulação e avaliação das redes de sensores sem fios que respeitam a norma IEEE 802.15.4 e que possuem requisitos de tempo real. No nosso trabalho incluímos todas as funções de gestão necessárias para suportar o uso de intervalos de tempo de acesso garantidos (GTS) para as transmissões de tramas.

A concretização destes mecanismos foi avaliada e validada através de casos de teste, utilizando diferentes cargas de rede e métricas de desempenho, tais como a taxa de entrega, latência e consumo de energia, permitindo uma melhor caracterização das redes IEEE 802.15.4 no suporte de comunicações em tempo real.

Depois de melhorar o suporte do NS-2 para a avaliação de redes IEEE 802.15.4 com requisitos de tempo real, novos recursos foram necessários para complementar o modelo de erro atual do simulador NS-2, e para permitir a avaliação da rede sob condições de erro e mais especificamente eventos de inacessibilidade da rede.

Para validar o nosso estudo, desenvolveu-se uma ferramenta flexível, capaz de recriar os eventos de inacessibilidade e simular diferentes condições de erro na rede, apelidada de injetor de faltas. O modelo de erro atual presente no simulador não permite afetar uma trama específica, como uma trama MAC por exemplo.

Por isso, desenvolvemos um novo módulo para injetar falhas e analisar o comportamento da rede sob condições de erro. Como resultado geral, este trabalho tem o compromisso de estabelecer uma plataforma robusta de estudo de forma a proporcionar uma

melhor compreensão dos aspetos temporais das redes IEEE 802.15.4.

Assim o nosso principal objetivo neste trabalho é através de simulação validar os resultados obtidos no anterior estudo teórico sobre a inacessibilidade em redes de comunicações sem fios IEEE 802.15.4 pelo melhoramento das ferramentas de simulação existentes.

As principais contribuições do trabalho descrito nesta tese incluem:

- Avaliar o simulador de rede NS-2 identificando as suas limitações
- Melhoramentos no módulo IEEE 802.15.4 do NS-2 de forma a proporcionar um melhor suporte a simulação de redes com requisitos de tempo real.
- Incorporação dos mecanismos de CFP e GTS, através da implementação do GTS dentro do módulo IEEE 802.15.4 presente no NS-2.
- Desenvolver as operações de gestão normalizadas não concretizadas no modulo IEEE 802.15.4 presente na versão oficial do NS-2.
- Adição de novos recursos necessarios para a avaliação da rede em condições de erro, mais especificamente, um injetor de faltas, e um módulo de contabilização temporal e energético.
- A utilização destes recursos na validação de modelos teóricos existentes respeitantes à avaliação da inacessibilidade em redes IEEE 802.15.4.

Foram produzidos vários artigos no âmbito deste trabalho, alguns deles apresentando um trabalho preliminar sobre o assunto abordado, e os restantes resultantes do trabalho aqui descrito. Os seguintes documentos foram publicados em congressos nacionais:

- *Jeferson L. R. Souza, André Guerreiro, José Rufino, "Characterizing Inaccessibility in IEEE 802.15.4 Through Theoretical Models and Simulation Tools", em INForum 2012 - Simpósio de Informática, Lisboa, Portugal, Set. 2012.*
- *André Guerreiro, Jeferson L. R. Souza, José Rufino, "Improving NS-2 Network Simulator for IEEE 802.15.4 standard operation ", em INForum 2013 - Simpósio de Informática, Évora, Portugal, Set. 2013.*
- *André Guerreiro, Jeferson L. R. Souza, José Rufino, "Improving NS-2 Network Simulator to evaluate IEEE 802.15.4 wireless networks under error conditions", em SENSORNETS 2014 - International Conference on Sensor Networks, Lisboa, Portugal, Jan. 2014.*

Contents

List of Figures

List of Tables

Abbreviations

BI beacon interval.

CAP Contention Access Period.

CBR Constant Bit Rate traffic.

CFP Contention Free Period.

CSMA/CA Carrier Sense Multiple Access with Collision Avoidance.

DFDR Data Frame Delivery Ratio.

GTS Guaranteed Time Slot.

LLC Logical Link Control.

MAC Medium Access Control.

NAM Network Animator.

OTcl Object-oriented Tool Command Language.

PHY Physical layer.

RTS/CTS Request to Send / Clear to Send.

SSCS Service Specific Convergence Sub-layer.

SWaP size, weight, and power consumption.

WnS Wireless network Segment.

WSAN Wireless Sensor and Actuator Network.

WSN Wireless Sensor Network.

Chapter 1

Introduction

1.1 Motivation

Wireless networks technology are seen as the future of communications. Mobility, size, weight, and power consumption (SWaP), and the absence of cable infrastructure are some fundamental advantages of wireless communications. Due to their unique features, there is a huge interest in developing applications that use Wireless Sensor Networks (WSNs) and Wireless Sensor and Actuator Networks (WSANs) in different sectors such as natural resources monitoring [19], aerospace [34], vehicular [8], and industrial [30] as illustrated in figure 1.1. Most of these environments have real-time communication constraints, which implies that the WSNs and WSANs must be capable to provide support on real-time communication services and provide guarantees about transmission time bounds.

However, the open and shared communication medium used by wireless networks is highly susceptible to electromagnetic interferences, and obstacles on the communication path, which may disturb the communications performed by the Medium Access Control (MAC) layer. Improving the timeliness and dependability of MAC level services is of utmost importance to provide a real-time data transmission service on wireless communications.

There are many studies in wireless communications focused in the provision of reliable and real-time communication services at the lowest level of the protocol stack [9, 29, 10]. However these studies pay little or no attention to the dependability aspects of MAC sublayer and its services, which are essential to assure the timeliness and resilience of the network when operating under error conditions.

Such faults may affect the MAC layer operation itself and induce temporary network partitioning, dubbed network inaccessibility [39], which imposes impairments fulfilling network operation with real-time properties. A previous theoretical study [31] indicates that the occurrence of network inaccessibility may be a source of transmission protocol delays, which may induce application deadline misses that may compromise the depend-

Figure 1.1: WSN and WSAN Real-Time Applications

ability and timeliness properties of the whole networked system.

This study represents a motivation for the current research on the test and evaluation of WSNs and WSANs through network simulators, where we take the IEEE 802.15.4 network standard and his potential to support real-time traffic (through the allocation of exclusive network access) as a case study.

The use of network simulators is a suitable tool to test and evaluate network behaviours using different environmental conditions. There are several network simulators available [16], some with commercial license, such as OPNET [24], and others with open source or academic license, like OMNeT++ [23], Prowler [26], TOSSIM [18], and NS-2 [21]. The NS-2 simulator is the most accepted and widely used network simulation tool on the literature, being open source and modular, supporting the simulation of WSNs and WSANs through the IEEE 802.15.4 standard [11].

Although NS-2 has an IEEE 802.15.4 module [41], this module does not have a native support for features that address real-time aspects of communications, such as emulation of a Contention Free Period (CFP) where time slots can be allocated for exclusive access to the network. One objective is to overcome the existing limitation which, natively, only allow contention-based communications in the IEEE 802.15.4 NS-2 module.

Therefore, this work presents improvements in the IEEE 802.15.4 NS-2 module to provide a better support for the test, simulation and evaluation of IEEE 802.15.4 networks with real-time requirements. We include all the management functions needed to support

the use of Guaranteed Time Slot (GTS) for frame transmissions, adapting and extending an implementation of a CFP module proposed by [5]. We evaluate and validate our implementation through test cases that uses different network loads, and performance metrics such as delivery ratio, latency, and energy consumption, allowing a better characterization of IEEE 802.15.4 networks in the support of real-time communications.

After enhancing the NS-2 support to the evaluation of IEEE 802.15.4 networks with real-time requirements, new features are needed to complement the current NS-2 error model, and allow the evaluation of the network under error conditions and more specifically network inaccessibility events.

To validate our study we developed a flexible tool capable of re-create the inaccessibility events and simulate different error conditions on the network, dubbed fault injector. The current error model cannot affect a specific frame such as MAC. So we developed a new module to inject faults and analyse the network behaviour under error conditions.

As a overall result, this work is committed to establish a robust study platform to provide a better understand of the temporal aspects of IEEE 802.15.4.

1.2 Objectives

The main goal of this work is through simulation validate the results obtained in the previous theoretical study about network inaccessibility in IEEE 802.15.4 wireless communications by enhance the simulation tools. Thus, this work addresses:

- Complement the IEEE 802.15.4 NS-2 module with CFP to support features that address real-time aspects of communications.
- Enrich the network simulator (NS-2) to measure network inaccessibility on a simulation environment.
- The validation of the previous theoretical study about network inaccessibility in IEEE 802.15.4 wireless communications.
- The extraction of real-time metrics from the comparison of results obtained by the theoretical study and simulation experiments.
- Analyse the impact of network inaccessibility in the power consumption of the wireless device.

1.3 Contributions

The main contributions of the work described in this thesis comprise:

- Evaluate the network simulator NS-2 to identify its limitations

- Improvements in the IEEE 802.15.4 NS-2 module to provide a better support for the emulation of networks with real-time requirements.

- Incorporation of the CFP and GTS mechanisms, through GTS definition within the IEEE 802.15.4 module present in the NS-2.

- Develop IEEE 802.15.4 standard management operations not implemented in the official NS-2 release.

- Add new features to complement the current NS-2 error model, and allow the evaluation of the network under error conditions and more specifically network inaccessibility events.

- These new features include a tool capable of corrupt specific frames, dubbed fault injector, and a temporal and energetic account module.

1.4 Institutional context

The development of this thesis took place at the Navigators team in Large-Scale Informatics Systems Laboratory (LaSIGE-FCUL), a research unit of the Informatics Department (DI) of the University of Lisbon, Faculty of Sciences. This work was developed within the scope of the FP7 Project KARYON (Kernel-Based ARchitecture for safetY-critical cONtrol), granted to the Timeliness and Adaptation in Dependable Systems research line of the Navigators group. The author of this thesis integrated the Navigators KARYON team as a junior researcher.

1.5 Publications

There were produced several articles in the scope of the KARYON project, some of them presenting preliminary work on the subject approached in this thesis, and the remaining resulting of the work herein described. The following papers were published in national conferences:

- *Jeferson L. R. Souza, André Guerreiro, José Rufino, "Characterizing Inaccessibility in IEEE 802.15.4 Through Theoretical Models and Simulation Tools", in INForum 2012 - Simpósio de Informática, Lisbon, Portugal, Sept. 2012.*

- *André Guerreiro, Jeferson L. R. Souza, José Rufino, "Improving NS-2 Network Simulator for IEEE 802.15.4 standard operation ", in INForum 2013 - Simpósio de Informática, Évora, Portugal, Sept. 2013.*

- *André Guerreiro, Jeferson L. R. Souza, José Rufino, "Improving NS-2 Network Simulator to evaluate IEEE 802.15.4 wireless networks under error conditions", in SENSORNETS 2014 - International Conference on Sensor Networks, Lisbon, Portugal, Jan. 2014.*

1.6 Document structure

To present the contributions of this work, the document is organized as follows: The **Chapter 2** describes the important concepts in real-time communication and in wireless sensor networks technologies giving particular attention to the IEEE 802.15.4 standard and his potential to support real-time traffic through GTS mechanism, as well as the most relevant simulations tools available for WSN. At **Chapter 3** will be presented the effect of network inaccessibility in Wireless Sensor Networks, characteristics and definitions, which is the foundation of this work. **Chapter 4** defines the challenges addressed in this thesis such as improving the IEEE 802.15.4 NS-2 simulation module for real-time operation support, and presents the evaluation results of different real-time metrics performed on IEEE 802.15.4. **Chapter 5** addresses the evaluation of inaccessibility scenarios through fault injection, presenting an fault injector that allows to simulate accidental errors on the network operation. A temporal and energetic analysis under error conditions is conducted and finally network inaccessibility results are presented comparing simulated and theoretical values. Finally **Chapter 6** shows some concluding remarks of the work approached in this thesis and highlights future work developments.

Chapter 2

Background Context

This chapter introduces fundamental concepts, an information background required to understand the issues addressed in this thesis. The chapter starts with a description of the IEEE 802.15.4 protocol which is one potential candidate to achieve predictable real-time support in WSNs and WSANs and an object of study in this work. Then we present a brief overview of the state of the art addressing the available tools to evaluate real-time communications on WSNs and WSANs.

2.1 The IEEE 802.15.4 Standard

The IEEE 802.15.4 specification [11] is a standard that allows the creation of wireless networks, being more specifically oriented for the creation of WSNs and WSANs. Each IEEE 802.15.4 network has a special node dubbed network coordinator, which defines a set of characteristics of the network such as addressing, supported channels, and operation mode.

Important features include node association, which is the service used to establish membership for a node in a network. Different network topologies (star and peer-to-peer) are available and real-time suitability by reservation of guaranteed time slots. Nodes also include power management functions such as link quality, used to indicate how strong the communications link is and energy detection which is a type of scan based on signal strength.

The network can operate either in a beacon-enabled mode or in a nonbeacon-enabled mode. In the beaconless mode, the protocol is essentially a simple Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol. Since most of the unique features of IEEE 802.15.4 are in the beacon-enabled mode, like support for communications with real-time restrictions we will focus our attention on this mode.

In the beacon-enabled mode the network coordinator manages the access to the network by periodically transmitting a special frame dubbed Beacon, which delimits the structure dubbed superframe, depicted in figure 2.1. The period between consecutive

beacon transmissions is dubbed beacon interval (BI).

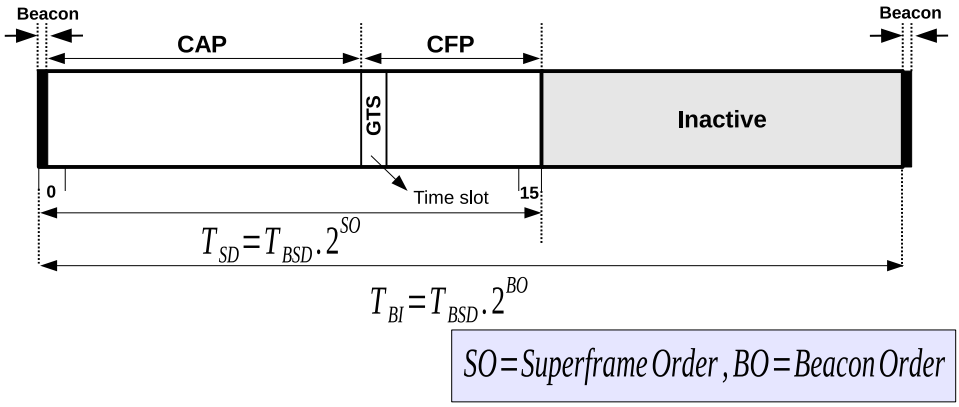

Figure 2.1: Superframe structure

There are both active and inactive portions in the superframe. Nodes communicate with their coordinator only during the active period and enter a low power mode during the inactive period. Constants and variables used for IEEE 802.15.4 network configuration and parametrisation are summarized in table 2.1. The parameter BO decides the length of beacon interval ($\mathcal{T}_{BI} = 2^{BO} \times \mathcal{T}_{BSD}$) and the parameter SO describes the length ($2^{SO} \times \mathcal{T}_{BSD}$) of the active portion of the superframe. The active portion of each superframe is further divided into 16 equal time slots and consists of three parts: the beacon, a Contention Access Period (CAP) and a CFP (which is only present if GTS slots are allocated by the coordinator to some of the node). Each GTS consists of some integer multiple of CFP slots and up to 7 GTS are allowed in CFP. The parameter BE is the backoff exponent, which is related to how many backoff periods a node shall wait before attempting to assess a channel.

IEEE 802.15.4 Name	**Abbr**
$aBaseSuperFrameDuration$	$\mathcal{T}_{BSD}$
$macBeaconOrder$	BO
$macSuperframeOrder$	SO
BI	$\mathcal{T}_{BI}$
$macMaxCSMABackoffs$	$maxBackoff$
Backoff Exponent	BE

Table 2.1: Relevant time-related constants of IEEE 802.15.4 Standard

We now present some key features of IEEE 802.15.4 MAC that will be addressed further on this work.

2.1.1 Frame Format

The standard defines four MAC frame types:

- A beacon frame, used by a coordinator to transmit beacons
- A data frame, used for all transfers of data
- An acknowledgement frame, used for confirming successful frame reception
- A MAC command frame, used for handling all MAC peer entity control transfers

To these MAC frame types correspond values of the frame type field, as presented in table 2.2.

Frame type value	Description
000	Beacon
001	Data
010	Acknowledgment
011	MAC command

Table 2.2: Values of the Frame Type field

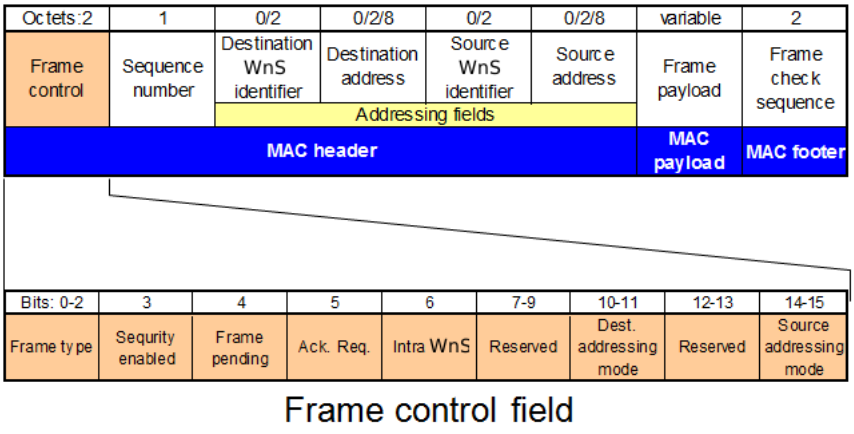

Figure 2.2: General MAC frame format and format of the Frame Control field

The general MAC frame format is represented in figure 2.2. The MAC header contains the information of MAC level (used in IEEE 802.15.4 frames). Is 9 bytes long and is composed by the Frame control field: 2 bytes, the Sequence number: 1 byte, Destination WnS address: 2 bytes, Destination address mode: 2 bytes, Source address mode: 2 bytes.

The MAC command frames defined by the MAC sublayer are listed in table 2.3. The **association request** command allows a node to request association with a WnS through

Command frame identifier	Command name
0x01	Association request
0x02	Association response
0x03	Disassociation notification
0x04	Data request
0x05	Coordinator conflict notification
0x06	Orphan Node notification
0x07	Beacon request
0x08	Coordinator realignment
0x09	GTS request

Table 2.3: MAC command frames

the coordinator. This command shall only be sent by an unassociated node that wishes to associate with a WnS.

The **association response** command allows the coordinator to communicate the results of an association attempt back to the node requesting association.

The coordinator or an associated node may send the **disassociate notification** command. The data request command is sent by a node to request data from the coordinator.

The **coordinator conflict** notification command is sent by a node to the coordinator when a coordinator conflict is detected. The **orphan node** notification command is used by an associated node that has lost synchronization with its coordinator. The **beacon request** command is used by a node to locate all coordinators within its radio communications range during an active scan.

The **coordinator realignment** command is sent by the coordinator following the reception of an orphan node notification command from a node that is recognized to be on its WnS. If this command is sent following the reception of an orphan node notification command, it is sent directly to the orphaned node. If this command is sent when any WnS configuration attributes (i.e., WnS network identifier, short address, channel, or channel page) change, it is broadcast to the WnS.

Finally the **GTS request** command is used by an associated node that is requesting the allocation of a new GTS or the deallocation of an existing GTS from the coordinator.

2.1.2 Contention Access Period (CAP)

The CAP starts right after the beacon and before the CFP on a superframe, and all frames in the CAP use slotted CSMA/CA. When a node needs to transmit during the CAP, it enables its receiver and delays for a random number of complete backoff periods (up to $2^{BE} - 1$ periods) and then determines if the channel is clear. A backoff period is a period where the node waits for a amount of time before attempting to retransmit. The

MAC ensures that, after the random backoff, the remaining CSMA/CA operations can be undertaken and the entire transaction can be completed before the end of CAP. A transaction represent the exchange of related, consecutive frames between two peer MAC entities, required for a successful transmission of a MAC command or data frame. If the channel is busy, the MAC delays for a random time and tries a number of times less than or equal to $macMaxCSMABackoffs$, otherwise it terminates with a failure.

2.1.3 Contention Free Period (CFP)

The IEEE 802.15.4 standard allows the optional use of CFP for nodes that require dedicated bandwidth to achieve low latencies. The CFP was designed to support real-time traffic, being divided in transmission windows dubbed GTSs that use an exclusive and contention-free approach in the access of the network. The CFP is defined in the superframe between the slot boundary immediately following the CAP and the start of the next beacon. All contention-based transactions are completed before the CFP begins. When a node wishes to transmit a frame using GTS, it first checks a list on the beacon frame to see whether it has been allocated a valid GTS. If a valid GTS is found, the node enables its receiver at a time prior to the start of the GTS and transmits the data during the GTS period. The MAC layer of the coordinator ensures that its receiver is enabled for all allocated guaranteed time slots. Once a given GTS slot is allocated to a node, only this node can transmit in this time interval.

2.2 A Survey of Simulators for Wireless Sensor Networks

As the technologies for wireless nodes improve, the requirements for networking are increasing. That enables possibilities for new applications. To reduce costs and time of the deployment process, simulation of the network is a preferred task before testing with real hardware. There are general purpose and specific WSN simulators [16], as listed in table 2.4. We address some of the most used and popular simulators [16, 12] giving more prominence to the selected one to perform this work, the NS-2.

	Prowler	OMNeT++	OPNET	NS-2	NS-3
802.15.4 support	802.15.4, fair (adhoc routing)	Not the whole standard	yes	yes	Not yet
Documentation	poor	yes	yes	yes	yes
License	academic	academic	commercial	GPL	GPL
User Friendly	Graphic UI	Graphic UI	Graphic UI	No Graphic UI	No Graphic UI

Table 2.4: Simulators comparison

Apart from the NS-2 there are other popular simulators such as NS-3 [22] which model node is thought more like a real computer and has a behaviour closer to it. NS-3 is intended to eventually replace the NS-2 simulator, however does not have yet support for the IEEE 802.15.4 standard.

OMNeT++ [12], is also a public source component-based discrete event network simulator, OMNeT++ is a very good software with a lot of documentation. Its graphical interface makes it more user friendly than others. However, only part of the 802.15.4 standard is implemented, therefore, it reduces its application.

Prowler [12] is an event-driven wireless network simulator designed to run in Matlab environment. OPNET [12] Modeler is a discrete event, object oriented, general purpose network simulator. OPNET have a very good documentation, graphic UI. However, the counterparts for using OPNET freely are too heavy and risky because there is no guarantee the license would be renewed.

There are many other simulators not mentioned in table 2.4, such as VisualSense [12] which is a component-based modeling and simulation framework built on Ptolemy II for wireless sensor networks, Castalia [12] which is a simulator for WSN, Body Area Networks and networks of low-power embedded devices.

Nevertheless the NS-2 was chosen given its modularity open source license enhanced features, support of real time simulation and is capable of model different kind of wireless and wired networks and protocols, so it represents a very useful tool to study the dynamics of a communication network under different types of scenarios.

2.3 NS-2 Network Simulator

The network simulator NS-2 is a discrete event simulator developed in a collaborative effort by many institutions, containing contributions from different researchers [20]. As a discrete-event simulator, all actions in NS-2 are associated with events rather than time. NS-2 was developed using two key languages: C++ and Object-oriented Tool Command Language (OTcl). While the C++ defines the internal mechanisms (i.e., a backend) of the simulation objects, the OTcl sets up simulation by assembling and configuring the objects as well as scheduling discrete events (i.e., a frontend). The C++ and the OTcl are linked together using TclCL as illustrated in the figure 2.3. Mapped to a C++ object, variables in the OTcl domains are sometimes referred to as handles. Conceptually, a handle (e.g., n as a node handle) is just a string in the OTcl domain, and does not contain any functionality. Instead, the functionality (e.g., receiving a packet) is defined in the mapped C++ object (e.g., of class Connector). In the OTcl domain, a handle acts as a frontend which interacts with users and other OTcl objects. It is possible to define its own procedures and variables to facilitate the interaction. The member procedures and variables in the OTcl domain are

called instance procedures. NS-2 also uses a Network Animator (NAM). It is a Tcl/TK based animation tool for viewing network simulation traces and real world packet traces.

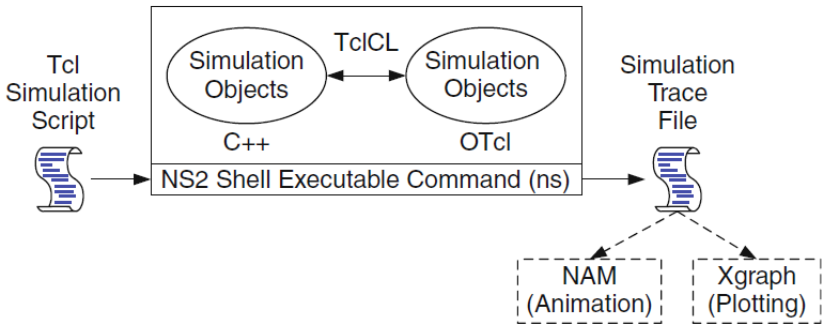

Figure 2.3: NS-2 Architecture [13]

2.3.1 NS-2 Error Model

An error model is an NS-2 module which imposes error on packet transmission. Derived from class Connector, it can be inserted between two NsObjects. An error model simulates packet error upon receiving a packet. If the packet is simulated to be in error, the error model will either drop the packet or mark the packet with an error flag. If the packet is simulated not to be in error, on the other hand, the error model will forward the packet to its downstream object. An error model can be used for both wired and wireless networks.

In the current version of the simulator, the error model is implemented to simulate the errors by either marking the frames with error flags or dumping the frames to a drop target. If the drop target exists, it will received corrupted packets from ErrorModel. Otherwise, ErrorModel just marks the error flag of the packets common header, thereby, allowing the upper NsObject to handle the loss. To add an error model over wireless networks, each node can insert a given statistical error model either over outgoing or incoming wireless channels.

In the implementations, the unit of error can be specified in terms of frames, bits or time-based to support a wide variety of models such as: ErrorModel/Trace which is a error model that reads a loss trace (instead of a math/computed model); ErrorModel/Periodic: models periodic packet drops (drop every nth packet we see); SelectErrorModel: for Selective packet drop; ErrorModel/TwoState: Two-State: error-free and error; ErrorModel/List: specify a list of packets/bytes to drop, which could be in any order;

Nevertheless, none of this models represented in figure 2.4 are capable of mark a specific frame to drop, as MAC control frame, a beacon frame for example. In essence, all these models are completely useless for the study of network inaccessibility.

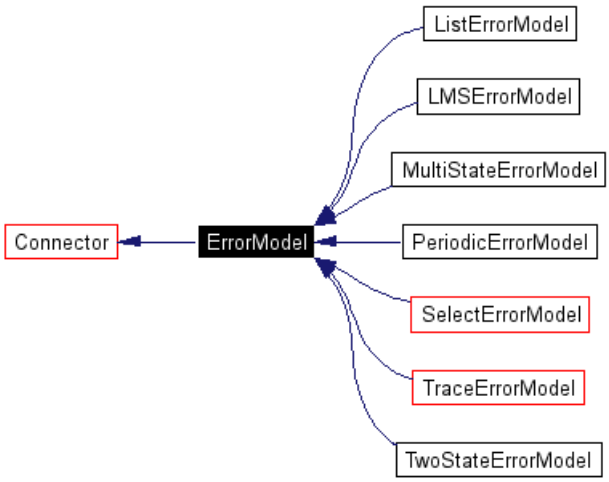

Figure 2.4: NS-2 Error Models.

2.3.2 NS-2 Energy Model

The energy model represents the level of energy in a wireless node. There is only a single class variable $energy$ which represents the level of energy in the node at any given time. The energy model in a node has a initial value which is the level of energy the node has at the beginning of the simulation. This is known as $initialEnergy$. The constructor EnergyModel($initialEnergy$) requires the initial-energy to be passed along as a parameter. It also has a given energy usage for every frame it transmits and receives. These are called $txPower$ and $rxPower$. These parameters units are represented in table 2.5 and the default values defined by the NS-2 developers. When the energy level at the node goes down to zero, the value in $energy$ variable, no more packets can be received or transmitted by the node. The energy model in NS-2 only models the power consumed by the

Attribute	Optional values	Default values
$rxPower$	receiving power in watts (e.g 0.3)	281.8mW
$txPower$	transmitting power in watts (e.g 0.4)	281.8mW
$initialEnergy$	energy in joules (e.g 0.1)	0.0

Table 2.5: Parameters for the energy model configuration

transceiver, and does not include the micro-controller.

2.3.3 IEEE 802.15.4 NS-2 Simulator Module

Within the NS-2 simulation modules the IEEE 802.15.4 NS-2 module which is provided in the form of methods of each layer class specified in the IEEE 802.15.4 standard [11] and which the module architecture is represented in figure 2.5. The Service Specific Con-

vergence Sub-layer (SSCS) is the interface between MAC and the Logical Link Control (LLC). It provides a way to access all the MAC primitives, but it can also serve as a wrapper of those primitives for convenient operations. It is an implementation specific module and its function should be tailored to the requirements of specific applications.

However the communication during CFP is not implemented in a modular way in the current IEEE 802.15.4 NS-2 module. The absence of the GTS mechanism is a major drawback once is fundamental for real-time WSN and WSAN applications, allowing a node to operate on the channel within a portion of the superframe that is dedicated exclusively to it.

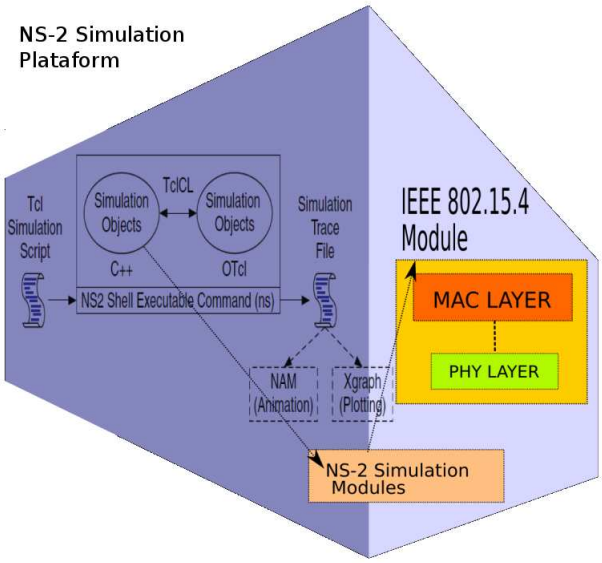

Figure 2.5: NS-2 IEEE 802.15.4 module architecture.

We identified some differences in the behaviour regarding the implementation of the standard MAC management actions on the current IEEE 802.15.4 module off the NS-2 simulator.

For example, the backoff calculation of the CSMA/CA algorithm used by IEEE 802.15.4 uses a uniform distribution. The number of backoff slots is chosen randomly, however, the random sequence is the same for every simulation. That is, if we run more than once the same simulation, we will have exactly the same results. On the current version of NS-2 the number of available channels to perform a scan is limited to 3.

Actions needed for the support of real-time data transmissions such as GTS allocation and deallocation are not implemented. Additionally, other management actions, and all auxiliary mechanisms needed to support the execution of such actions lack of implementation and should be implemented and incorporated in the IEEE 802.15.4 module, enhancing the compliance with the IEEE 802.15.4 standard.

2.4 Summary

The IEEE 802.15.4 standard is a potential candidate to support the creation of WSNs and WSANs that can be used in applications with real-time constrains. Here we are focused on the beacon-enabled mode operation of the network, designed to support data transmissions with temporal restrictions, and which is the target mode of our analysis and simulations.

The information inside the beacon helps the nodes to know the entire duration of the superframe, allowing the synchronization and the control of the medium access.

If a glitch in the medium occurs and a node does not receive the beacon frame is lost, the node stays inaccessible until the next beacon reception. These periods are much higher than a data frame loss and can jeopardize the normal operation of the network. The solution to minimize the problems caused by the occurrence of inaccessibility periods is to define means to control the inaccessibility. This control is based on the knowledge of all inaccessibility scenarios present in the network which strengthens the importance of developing tools to get this knowledge.

Different simulation tools have been addressed, however the NS-2 simulator was chosen for our validations. An overview regarding the different components and indicating some limitations of this tool was conducted, exposing the need to improve this tool in order to provide better support to the evaluation of WSN.

Chapter 3

Inaccessibility in Wireless Sensor Networks

3.1 Introduction

WSNs and WSANs has seen as the network infrastructure of the future. The main advantage of wireless networks is the flexibility provided by non-existence of cables and the reduced SWaP of the devices.

However disturbances induced in the operation of MAC protocols may create temporary partitions in the network, derived of the time required to detect and recover from these situations. These disturbances can be produced by external interferences or by some glitches in the operation of the MAC sub-layer. A solution for controlling these partitions in LAN-based networks was presented in [39]. These temporary network partitions are called inaccessibility [28, 37] and the definition of this concept is summarized here:

> *Certain kinds of components may temporarily refrain from providing service, without that having to be necessarily considered a failure. That state is called inaccessibility. It can be made known to the users of network components; limits are specified (duration, rate); violation of those limits implies permanent failure of the component.*

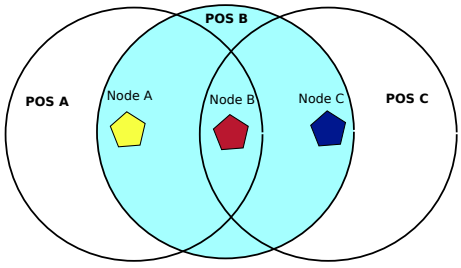

Figure 3.1: Hidden Node Problem

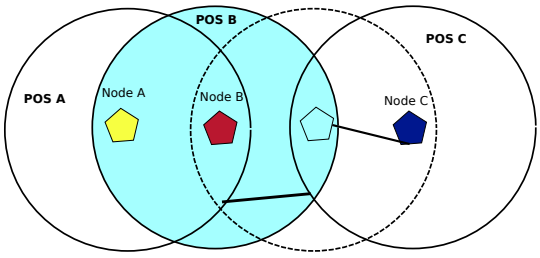

Figure 3.2: Mobile Node Problem

The problem of inaccessibility in wireless networks has been introduced in [31, 32]. In wireless networks, network inaccessibility events may be originated externally or derived of the proximity and position of a node, in relation to operating space of other nodes. The circles in figure 3.1 show the transmission and interference range of three different nodes. In the example presented in figure 3.1, the node A may overlap, total or partial, the frame transmission of node B and vice-verse. It may result in periods of inaccessibility for the two nodes. In a wireless network a hidden node refer to a node that is out of range of other nodes or a collection of nodes. The Request to Send / Clear to Send (RTS/CTS) handshake used in IEEE 802.11 tries to solve the hidden node problem. However, this technique does not solve completely the problem and increases the overhead of a transmission, an unacceptable condition, for example, for wireless sensor networks [15]. The node mobility, provided by wireless technology, allows the change of a node location easily. This mobility may cause connection loss between nodes. Figure 3.2 shows that, after moving, node C is outside of node B range and it may cause periods of inaccessibility in both nodes. An environment with a high level of node mobility may cause the occurrence of various periods of inaccessibility if the nodes move constantly their position to outside of each other range. The inaccessibility time, in both cases, is the time a node needs to re-establish normal operation of the MAC protocol. The knowledge of inaccessibility time bounds is important to achieve the support of real-time communication over wireless networks.

To minimize the problems caused by the occurrence of periods of network inaccessibility is important to define means to control the network inaccessibility. This work aims to validate the previous theoretical study about network inaccessibility in IEEE 802.15.4 wireless communications, using network simulations. On a first phase enrich the NS-2 to measure network inaccessibility on a simulation environment. Finally to compare the results obtained by the two different approaches, theoretical and simulation.

3.2 Preliminary Work

This section provides a brief explanation of network inaccessibility in wireless sensor networks, as well as a summary of the study [31, 14] to be validated.

3.2.1 System Model

The system model is formed by a set of wireless nodes[1] $X = \{x_1, x_2, \ldots, x_n\}$, being $1 < n \leq \#A$, where A is the set of all wireless nodes using the same communication channel. Figure 3.3 presents a graphical representation of X, which is supported by the following assumptions:

[1] A wireless node is a networked device capable to communicate with other nodes

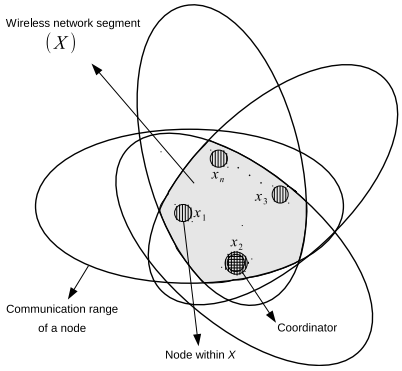

Figure 3.3: The graphical representation of a wireless network segment.

1. The communication range of X, i.e. its broadcast domain, is given by: $B_X = \bigcap_{j=1}^{n} B_D(x),\ \forall x \in X$, where $B_D(x)$ represents the communication range of a node x;

2. $\forall x \in A, x \in X \iff B_D(x) \bigcap B_X = B_X$ or, as a consequence of node mobility, $x \notin X \iff B_D(x) \bigcap B_X \neq B_X$;

3. $\forall x \in X$ can sense the transmissions of one another;

4. $\exists x \in X$ which is the coordinator, being unique and with responsibility to manage the set;

5. A network component either behaves correctly or crashes upon exceeding a given number of consecutive omissions (the component's *omission degree*, f_o) in a time interval of reference[2], $\mathcal{T}_{rd}$;

6. failure bursts never affect more than f_o transmissions in a time interval of reference, $\mathcal{T}_{rd}$;

7. omission failures may be inconsistent (i.e., not observed by all recipients).

The set X itself represents a network entity dubbed Wireless network Segment (WnS), as depicted in Figure 3.3. For a given WnS, assumptions 1, 2, and 3 define the physical relationship between nodes, assumption 4 defines the existence of a coordinator, and assumptions 5, 6, and 7 define how communication errors within the WnS are handled. All communications and relations between nodes are established at MAC level, which are reinforced by assumption 3. As a consequence of mobility, nodes may be driven away of a given WnS (assumption 2). All communication errors within WnS are transformed into

[2]For instance, the duration of a given protocol execution. Note that this assumption is concerned with the total number of failures of possibly different nodes.

omissions (assumption 5), and in the context of network components an omission is an error that destroys a data or control frame.

3.2.2 Network inaccessibility

There are two different types of frames that can be affected by the occurrence of disturbances on the normal network operation, control and data frames. The first one is used to manage and sustain the network operational. Whenever the control frame transmissions are corrupted with errors, the MAC layer has to execute actions in order to maintain the network operation after the occurrence of these errors. This is called a period of network inaccessibility, which is the time interval between the moment that the errors mentioned above occurs, and the normal network operation is restored. A node, during the referred period, is unable to access the network, and cannot communicate with other nodes. Due to this a temporary blackout on the network communication services occurs.

3.2.3 Theoretical modeling of network inaccessibility in IEEE 802.15.4

In Table 3.1 we present a collection of easy-to-use formulas defining the durations of periods of network inaccessibility. The worst case duration, (represented by the superscript ^{wc}, presented for each network inaccessibility scenario. The different parameters used in the formulas of Table 3.1 are as follows: $nrchannels$, represents the number of channels to be scanned; $nrWait$, defines the waiting period for a beacon frame in each channel scan, assuming the default value of $nrWait = 32$ in the IEEE 802.15.4 standard; $\mathcal{T}_{MAC\_ack}(frame)$ and $\mathcal{T}_{MAC\_ack}(frame)$ represent the delay from request to confirmation of a MAC frame transmission time with and without acknowledgement, respectively; $\mathcal{T}_{MLA}(action)$ represents the time needed to perform the specified action at the MAC management sublayer. Without loss of generality, an uniform value of $\mathcal{T}_{MLA}(action) = \mathcal{T}_{BI}/10$ is assumed for the duration of each MAC management sublayer action.

For the relevant scenarios, we describe next how the corresponding periods of network inaccessibility are obtained. The beacon frame controls the access to the network, and its reception is essential to maintain all the nodes synchronized within the different periods of the superframe structure. If a beacon frame is not correctly received an inaccessibility incident occurs. Thus, a **beacon frame loss** occurs when only one beacon is lost. The value of this period of inaccessibility is $\mathcal{T}_{BI}$ plus one $\mathcal{T}_{BSD}$ period, which is utilized as a margin to overcome some clock deviations that may occur between nodes.

The **multiple beacon frame loss** occurs when multiple and consecutive beacons are lost and a correct beacon frame is successfully received after the loss of $nrLost$ beacons. The **synchronization loss** is a special case of the multiple beacon frame loss scenario where after the loss of $nrLost$ beacons the next beacon is also lost.

Scenario	Equation
Single Beacon Frame Loss	$\mathcal{T}^{wc}_{ina\leftarrow sbfl} = \mathcal{T}_{BSD} \cdot (2^{BO} + 1)$
Multiple Beacon Frame Loss	$\mathcal{T}^{wc}_{ina\leftarrow mbfl} = \mathcal{T}_{BSD} \cdot \left(2^{BO} + 1\right) \cdot nrLost$
Synchronization Loss	$\mathcal{T}_{ina\leftarrow nosync} = \mathcal{T}_{BSD} \cdot \left(2^{BO} + 1\right) \cdot nrLost$
Orphan Node	$\mathcal{T}^{wc}_{ina\leftarrow orphan} = \mathcal{T}_{ina\leftarrow nosync} + \mathcal{T}_{MLA}(Orphan)$ $+ \sum_{j=1}^{nrchannels} \left(\mathcal{T}^{wc}_{MAC}(Orphan) + nrWait \cdot \mathcal{T}_{BSD}\right) + \mathcal{T}^{wc}_{MAC\_ack}(Realign)$
Coordinating Orphan Realignment	$\mathcal{T}^{wc}_{ina\leftarrow realign} = \mathcal{T}_{MLA}(Realign) + \mathcal{T}^{wc}_{MAC\_ack}(Realign)$
Coordinator Conflict Detection	$\mathcal{T}^{wc}_{ina\leftarrow C\_Detection} = \mathcal{T}^{wc}_{MAC\_ack}(C\_Conflict)$
Coordinator Conflict Resolution	$\mathcal{T}^{wc}_{ina\leftarrow C\_Resolution} = \mathcal{T}_{MLA}(Conflict) + \sum_{j=1}^{nrchannels} \left[\mathcal{T}^{wc}_{MAC}(Beacon\_R) + nrWait.\mathcal{T}_{BSD}\right]$ $+\mathcal{T}_{MLA}(Realign) + \mathcal{T}^{wc}_{MAC}(Realign)$
Extract Request	$\mathcal{T}^{wc}_{ina\leftarrow extReq} = \mathcal{T}^{wc}_{MAC_{ack}}(ExtReq) + \mathcal{T}_{wait}$
GTS request	$\mathcal{T}^{wc}_{ina\leftarrow GTS} = \mathcal{T}^{wc}_{MAC\_ack}(GTS)$
Association	$\mathcal{T}^{wc}_{ina\leftarrow assoc} = \sum_{j=1}^{nrchannels} \left[\mathcal{T}^{wc}_{MAC}(Beacon\_R) + nrWait.\mathcal{T}_{BSD}\right] + \mathcal{T}_{MLA}(Beacon) + \mathcal{T}^{wc}_{ina\leftarrow extReq} +$ $\mathcal{T}_{MLA}(AssocReq) + \mathcal{T}^{wc}_{MAC\_ack}(AssocReq)$
Re-Association	$\mathcal{T}^{wc}_{ina\leftarrow reAssoc} = \mathcal{T}_{ina\leftarrow nosync} + \mathcal{T}^{wc}_{ina\leftarrow assoc}$

Table 3.1: Easy-to-use formulas defining the durations of periods of network inaccessibility

To recover from such loss of synchronization two different strategies were identified in the standard specification [11]. Each individual node chooses the recovery strategy to be used. If some data/control frame was received during the last beacon interval, the node assumes an **orphan** status; otherwise, a **re-association** procedure should be carried out. In both recovery strategies, the node looks for a coordinator in the given set of channels. After the channel scan, a **coordinator realignment** or a **re-association** procedure is performed within the **orphan** and **re-association** scenarios, respectively.

In the execution of the **association** procedure, the channel scan is followed by a beacon processing action, the extract of control information, an association processing action and the actual association with the coordinator. The **re-association** and **association** procedures are quite equivalent. The **association** procedure is executed when a non-coordinator node has no information about its coordinator.

A **coordinator conflict** occurs when more than one coordinator is active within the same network. By default, each network has a unique identifier, $networkID$, which identifies the network uniquely and is used by the coordinator in beacon transmissions. If some other (possibly old) coordinator enters the broadcast domain, e.g., after moving away during a long period of time, the network may have two different coordinators transmitting beacons with the same $networkID$. To solve such conflict, the actual coordinator performs a search within a set of specified channels. If the coordinator does not

found other coordinator sending beacons with its own identifier after the scan in all channels, no further action is taken and the network becomes accessible again. Otherwise, a new identifier is selected and, if necessary, a MAC coordinator realignment command is broadcast. In table 3.1, this scenario is separated in two individual contributions: **coordinator conflict detection**, to be performed upon the detection of a coordinator conflict and its notification; a longer **coordinator conflict resolution** procedure, which includes the channel search procedure.

The final scenarios do include the procedure required for requesting the allocation of a GTS slot and the procedure to extract control information from the coordinator. Timeliness and dependability properties of the network may be compromised by the consequence of network inaccessibility. The existent simulation tools should be enhanced to include mechanisms capable to test MAC sublayer operation under error conditions, providing then accurate analysis regarding the temporal aspects of the network.

3.3 Summary

In this chapter we presented an overview of a previous theoretical study [31] that shows that errors affecting MAC sublayer management operations may lead occurrence of ”blackouts“ within IEEE 802.15.4 wireless communications, where the network remains inaccessible by a temporary period of time. This period is dubbed network inaccessibility, and its characterization involves the detailed study of the corresponding MAC protocol operation. A comprehensive set of scenarios leading to network inaccessibility is thoroughly discussed. Network inaccessibility has a strong negative impact in the temporal behaviour of IEEE 802.15.4 networks, being extremely important its characterization.

Chapter 4

Improving the IEEE 802.15.4 NS-2 simulation module for real-time operation

4.1 Problem Definition

The original NS-2 simulator IEEE 802.15.4 module is not fully compliant with the standard, regarding the behaviour and support of transmissions with real-time requirements.

Although NS-2 is extensively used in wireless sensor network simulations with extended libraries, from our analysis, we discovered several aspects of NS-2 operation that need to be improved to secure the provisioning analysis work of real-time guarantees. These improvements are two fold: developing IEEE 802.15.4 standard management operations not implemented in the official NS-2 release; Implementing the absent mechanisms such as the communication in CFP.

4.2 Incorporating and enhancing MAC Management actions according to the Standard

The NS-2 simulator module has some differences in the behaviour regarding the implementation of IEEE 802.15.4 standard MAC management actions. Taking this in consideration we added the functions presented in table 4.1. Some operations are implemented in the original module but they are not fully functional. We corrected them and implemented other additional operations, as needed.

MAC Manege-ment Action	IEE 802.15.4 Standard Behaviour	NS-2 Original Module	NS-2 implemented module
Orphan	A request is issued to the MAC layer to start an orphan scan recovery action	Not Functional	Operational
Coordinator Realignment	On the reception of Orphan notific. is required an acknowledged trans-mission of a realignment command	Not Functional	Operational
Coordinator conflict	If two coordinators establish a net-work with the same Network iden-tifier, a Coordinator conflict occurs	Not imple-mented	Implemented
Channels Available to Scan	16 channels on 2.4Ghz	Only the first 3 channels	16 channels
Scan Duration atribute	$\mathcal{T}_{BSD} \times (2^n + 1)$, where n is the value of the $ScanDuration$ parameter.	incorrect definition	in compliance with the standard
Network Information Base (NIB) attribute	The Management Entity checks to see if the NIB attribute is a MAC or a PHY layer attribute.	This verifi-cation is not performed	in compliance with the standard

Table 4.1: NS-2 IEEE 802.15.4 Module behaviour comparison

The coordinator conflict is one of those such operations. A coordinator conflict has two phases, the detection and resolution. The coordinator conflict detection occurs when more than one coordinator is active within the same network. By default, each network has an identifier, the $networkID$, which identifies the network uniquely and is used by the coordinator in beacon transmissions. If some other (possibly old) coordinator enters the network operational space, e.g., after having been away from some period of time, the network may have two different coordinators transmitting beacons with the same $networkID$.

The coordinator conflict resolution in turn will request the MAC layer to perform an active scan. This scan is realized in all available logical channels, however in the current version of NS-2 the number of available channels is limited to 3. If the protocol management entities decide that the node was orphaned, a request is issued to the MAC layer to start an orphan scan recovery action, over a specified set of logical channels. For each logical channel: a MAC orphan notification command is sent; as reply, a MAC realignment command from the previously associated coordinator, is awaited during a given period. Once such MAC command is received the node terminates the scan and the network becomes accessible.

At the coordinator point of view, the need to assist MAC layer management actions starts when a MAC orphan node notification is received. Upon processing by protocol management entities, the acknowledged transmission of a MAC realignment command is requested as described in table 4.1. Relatively to the channel scan process carried out by the different nodes, they should be able to scan all available channels. However, since we aim to simulate a network operating in 2.4 Ghz we removed the limitation of scanning only the first 3 channels. This limitation was removed and the scan of all the 16 channels defined by the standard is now allowed as illustrated in table 4.1. The duration of each scan was also incorrect, once its parametrization was not set in compliance with

the standard. Again, the issue was corrected, as inscribed in table 4.1

4.3 CFP and GTS implementation in NS-2

On IEEE 802.15.4 networks data can be transmitted in three ways: *Direct transmission*, which means that data is sent during the CAP; *Indirect transmission*, which is only available for coordinators. The data is placed on the indirect transmission queue and is sent during the CAP when polled. And finally *GTS transmission*, which requires that a node has to use a GTS slot to transmit its data. To allow this, a GTS slot is allocated to the node for the specified data transmission. Although a data transmission request can occur at

Algorithm 1 Transmission Data Frame using GTS

1: Begin.
2: $MAC.Data.Send.Request(data)$;
3: **when** allocated GTS is reached **do**
4: $MAC.Data.transmit(data)$;
5: **end when**
6: End.

any-time in a superframe, a data transmission request using a GTS is required to transmit the data only during the allocated GTS. Therefore, in our implementation described in Algorithm:1, it is checked if the data transmission request using, represented on line 2, a GTS is in the allocated GTS duration or not, as represent on line 3.

After a GTS allocation is checked at the beacon, a timer for the expiration is started at the allocated GTS starting slot, and the data is transmitted during the allocated GTS interval as represent on line 4. Since the procedure to check the remaining GTS time is also implemented, multiple data can be transmitted during a GTS, which complies with the IEEE 802.15.4 standard.

Algorithm 2 Coordinator processing a GTS request command

1: Begin.
2: $MAC.Mgmt.GTS.Request(node_{addr}, nr\_slots)$;
3: **if** nr_slots are available **then**
4: $MAC.Mgmt.GTS.allocate(node_{addr}, nr\_slots)$;
5: $MAC.Mgmt.GTS.updateGTSList(node_{addr})$;
6: **else**
7: $MAC.Mgmt.GTS.Response(slots\_not\_available)$;
8: **end if**
9: End.

When a coordinator receives a GTS request command from a node willing to transmit data, Algorithm:2 is executed by the coordinator. After checking if the node GTS slot is valid (line 3), which means the $node_{addr}$ is already known by the coordinator and nr_slots are available, the allocation is made (line 4).

If the operation is successfully concluded, the final CAP slot subfield of the superframe specification field of the beacon frame is updated as in line 5, and the updated beacon is sent. If all the GTS slots are occupied at the time, an information regarding $slots\_not\_available$ is sent to the node, as described in line 7. The information from the GTS allocation or deallocation is delivered in the next beacon frame to the nodes that sent the GTS request command, letting them know the result of the requesting process.

4.4 Design and implementation of the solution

In order to achieve a better real-time support from the IEEE 802.15.4 simulation module, we extend the existent module to provide the GTS mechanism for network nodes. The adaptation was made changing some main classes of the IEEE 802.15.4 module, as represented in figure 4.1.

In the p802_15_4mac class, the entity that represents the MAC layer, exists a variable $txOption$ which corresponds to the transmission options of the IEEE 802.15.4 standard. This is defined as a static variable and is responsible for defining the options of data transmission. This implies that every node in the simulation environment deliver data using the same transmission options. This was modified allowing each node decide if want to transmit during CFP or CAP. Also a new timer was developed in order to control the expiration of an allocated GTS slot.

The GTS related methods that were provided in form of declaration but not implemented in the native version of the IEEE 802.15.4 module were also added both in the class p802_15_4mac class and p802_15_4sscs, which represent the connection between the MAC and the LLC layer and provides a way to access all the MAC primitives. The p802_15_4sscs interface is modified so that the GTS bit of the $txOption$ can be set or reset in the Tcl file. By doing so, it is possible for a specified node to be selected to use a GTS, and even a GTS can be controlled to be used or not by setting or resetting the GTS transmission bit.

Even though some structs for the GTS field exist and their simple implementations are available in p802_15_4field class, these do not suffice for communication using the GTS. Such examples include the final CAP slot input field of the superframe specification used at the time of the beacon frame generation and the use of the GTS in the MCPS-DATA.request primitive used at the time of the data frame transmission.

So on the p802_15_4field class, auxiliary methods had to be implemented, to support the management of the GTS allocation/deallocation mechanisms. However the adaptation of the CFP implementation in [5], to the latest version of the simulator used in this study, revealed some issues and incompatibilities.

The referenced implementation proposed in [5] prevents the use of MAC management commands such as Orphan Notification and Coordinator Realignment. In our implemen-

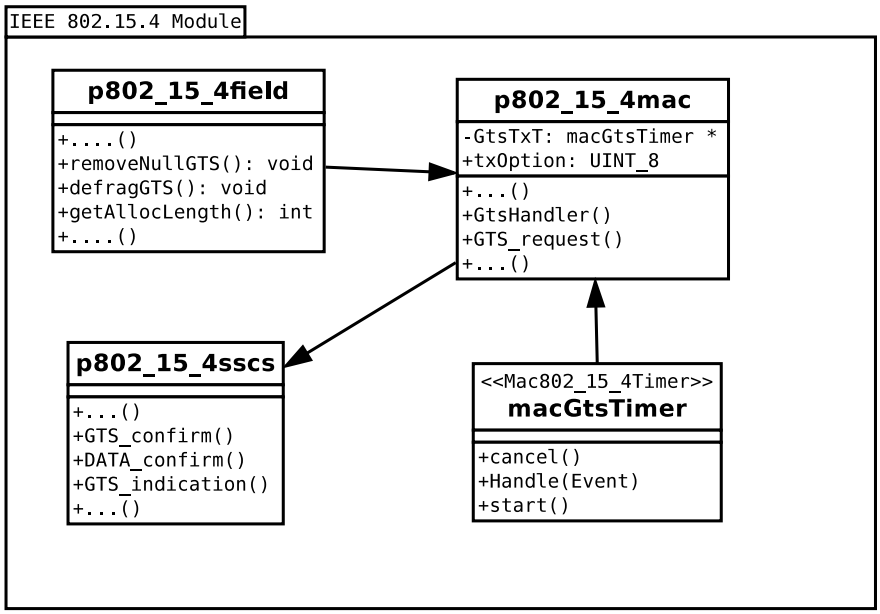

Figure 4.1: Class diagram of changed classes on native IEEE 802.15.4 module

tation this was corrected and the GTS can be activated from the NS-2 script without affecting other MAC services.

4.5 Data Analysis

In order to better understand the results of our simulations we developed some reporting tools, capable of summarize the important events that occur.

A performance report tool was also developed after defining some evaluation metrics described on the section 4.6. The tool was made through an *AWK* script that produces a report analysis regarding the defined metrics. *AWK* uses a data-driven scripting language consisting of a set of actions to be taken against textual data (either in files or data streams) for the purpose of producing formatted reports. Our script takes the NS-2 simulation log file as an input, and produce multiple text files with information about throughput, end-to-end delay, energy consumption, MAC control frames exchanged, traffic load and packet delivery ratio.

The NS-2 simulation log files are also used to generate a graphical analysis through a *gnuplot* script, giving a better knowledge of the events observed during simulations.

4.6 Evaluation metrics for an effective real-time communication in Wireless Sensor Networks

Several metrics can be defined to grade the performance of a technology against the elements of wireless networking. These metrics have been carefully chosen to demonstrate

the performance and the timeliness of the IEEE 802.15.4 networks. A detailed explanation of these metrics follows:

- *Data Frame Delivery Ratio (DFDR)*, which is the ratio between the total number of frames received in MAC sub-layer and the total number of data frame transmit requests during the simulation period. In our simulation we consider the data frames transmit requests issued by all the nodes but the coordinator.

$$DFDR = \frac{Total\ Data\ frames\ received \times 100}{Total\ Data\ frames\ transmit\ requests} \tag{4.1}$$

- *Latency*, which represents the transfer time of a data frame to a one-hop neighbour. For each individual data frame transfer, the frame transfer latency represents the interval between the instant when the data frame transmit request is issued $(Ttx\text{-}Data)$ and the instant of the corresponding data frame reception $(TrxData)$. This metric includes the data frame processing and queueing time at the nodes, the data frame transmission time and the back off interval (if applicable). The average latency can then be calculated over all successful end-to-end transmissions within the simulation run.

$$AverageLatency = \frac{\sum_{allreceivedframes}(TrxData - TtxData)}{Total\ number\ of\ received\ frames} \tag{4.2}$$

  On the other hand, the worst case value is given by:

$$WorstCaseLatency = \max_{allreceivedframes}(TrxData - TtxData) \tag{4.3}$$

- *Energy*, the energy model present in NS-2 is used to calculate the amount of energy consumed by the nodes during the simulation time.

$$Energy\ Used\ per\ Node = \frac{Total\ Energy\ Used}{Number\ of\ Nodes} \tag{4.4}$$

- *Throughput*, it measures the amount of data successfully received by the destination node within certain period of time.

$$Throughput = \frac{packets\_received \times packet\_size}{Simulation\ Time} \tag{4.5}$$

4.7 NS-2 Network Simulation Results

In this section we address a performance evaluation on different characteristics of IEEE 802.15.4 wireless networks. A comprehensive set of scenarios regarding the evaluation metrics, for an effective real-time communication, is thoroughly discussed.

Simulation Parameters	
NS-2 Version	2.35 updated with GTS features
Network Topology	Star Topology
Nodes	7
Traffic	Constant Bit Rate (CBR)
Reception range	15m
Carrier Sense range	15m
Packet Size	70 bytes
CAP Transmission Type	Direct, using CSMA/CA
CFP Transmission Type	GTS transmission
Transmission/Reception Power	30mW
Beacon	Enabled
Beacon Order	3
Superframe Order	3
Maximum CSMA/CA Attempts	4
Simulation Time	600 seconds

Table 4.2: Simulation Parameters

4.7.1 Simulation Setup

To evaluate our implementation, the simulations conducted in NS-2 will be analysed. First the appropriate evaluation metrics for an effective real-time communication in WSNs and WSANs are addressed in 4.6. Then is described the simulation set-up and finally the results achieved. Furthermore, all values were calculated and obtained based on a 2.4GHz IEEE 802.15.4 frequency operation.

The star topology network was chosen in this simulation. The network was simulated with seven nodes, where one of these nodes, in the center, was the coordinator. All other nodes are in the radio transmission range of the coordinator. Additionally all nodes are in a single broadcast domain, which means that all the nodes are within the range of each other. The simulation parameters are described in table 4.2.

To evaluate the network behaviour, the six remaining nodes constantly transmit data frames to the coordinator during CAP or CFP. The traffic generator is set to produce Constant Bit Rate traffic (CBR), which means data frames are transmitted at a constant rate from the nodes to the coordinator. The interval between each data transmission request is successively set to 1, 0.1, 0.01 and 0.001 seconds. Given the packet size of 70 bytes, this means the network load is monotonically increased, adopting the values of 0.07, 0.7, 7 and 70 KB/s. The MAC management actions required for node association with its coordinator and the GTS allocation times (if required) are excluded from the evaluation scope in the present simulation run.

4.7.2 Simulation Results

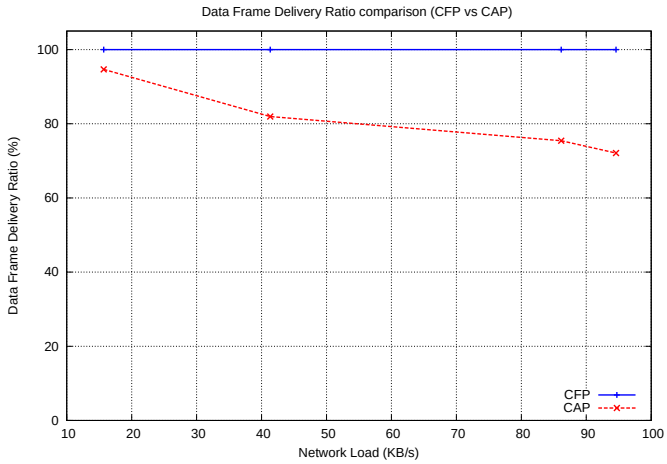

Figure 4.2: Data Frame Delivery Ratio comparison between transmission during CAP and CFP

Figure 4.2 represents the delivery ratio on the network, providing a comparison of the results achieved for transmission requests issued during the CAP and CFP periods. During CFP, nodes use the allocated GTS and get direct and exclusive network access, which allows to achieve about 100% delivery rate. In CAP the delivery ratio drops in function of the increase in the network load. This is explained by the occurrence of collisions during CAP, or due to the number of nodes attempting to access the medium. In the CSMA/CA protocol a data frame transmit request is dropped, if the number of transmission attempts exceed a given threshold defined by the Maximum CSMA/CA attempts (Table 4.2). This value represents the maximum number of backoffs the CSMA/CA algorithm will attempt before declaring a channel access failure.

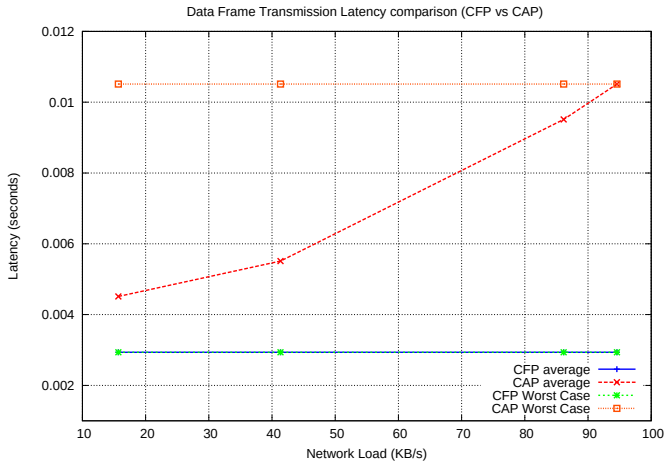

Figure 4.3: Data frame transmission Latency comparison between transmission during CAP and CFP

Figure 4.3 shows the latency comparison between a data frame transmission using CFP and CAP. While the latency remains almost constant when data frames are transmitted during CFP, using allocated GTS, the latency highly increases while using CAP. The constancy achieved in data frame transfer times during CFP is a sign of determinism and predictability and shows in figure 4.3 in two ways: an (almost) constant worst-case data frame transmission latency; the optimal value of this latency, which does not exceed 0.002936 seconds. This is due to nodes during CFP get exclusive network access, meaning nodes do not have to check if the media is idle and no collisions occur for those nodes. These results show the importance of the GTS mechanism in applications with real-time requirements on which deterministic data frame transmission times are mandatory. Additionally, the data frame transmission latency increases in CAP, up to the worst-case value of 0.010512 seconds, given the worst-case network load in the simulation setup.

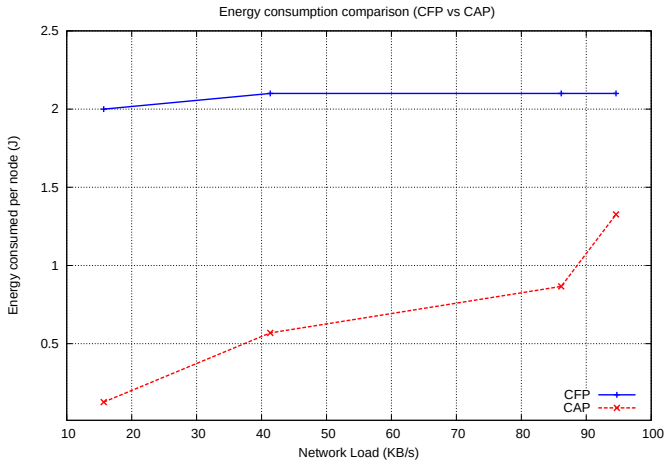

Figure 4.4: Energy consumption per node for data transmission during CAP and CFP

Finally, figure 4.4 represents the average energy consumption by the nodes during the simulation period. It worth noticing that the energy consumption increases when using CFP in comparison with CAP as result of required beacon frame reception tracking by the node, an action that obliges the node to switch-on its transceiver during the active period of every superframe instance. Contention-based access is more efficient under light network loads, whereas contention-free access becomes preferable when the background network load increases.

4.8 Summary

This chapter described improvements made to the IEEE 802.15.4 NS-2 simulator module, in the course of this thesis work. We identified limitations, and the absence of vital mechanisms required to support real-time simulations on WSN.

Therefore the support of NS-2 simulator to the IEEE 802.15.4 was enhanced with

addition of GTS mechanism and the unimplemented MAC management functions such as Coordinator Conflict, Orphan Node, Coordinator realignment.

Based on NS-2 simulations, we evaluate the performance of various features in the IEEE 802.15.4 MAC. We find that data transmission during the CAP reduces energy cost due to idle listening in the backoff period but increases the collision at higher rate and larger number of sources. While the use of GTS in the CFP can allow dedicated bandwidth to a device to ensure low latency, the device need to track the beacon frames in this mode, which increases the energy cost. The addition of available channels to scan during association revealed an increase of the association time an energy cost, but made the NS-2 more compliant to the standard.

Chapter 5

Evaluating Inaccessibility Scenarios through Fault Injection

5.1 Problem Definition

Given the lack of research related to network inaccessibility in wireless sensor networks, in the particular case of the standard IEEE 802.15.4, is important to have tools capable of analysing the impact of network inaccessibility on the network behaviour.

The existent simulation tools (Korkalainen et al., 2009) are not suitable to test and evaluate the behaviour of MAC sublayer services under error conditions, needing additional mechanisms to measure the temporal characteristics of MAC sublayer operation. With the purpose of validate the previous theoretical study about network inaccessibility,

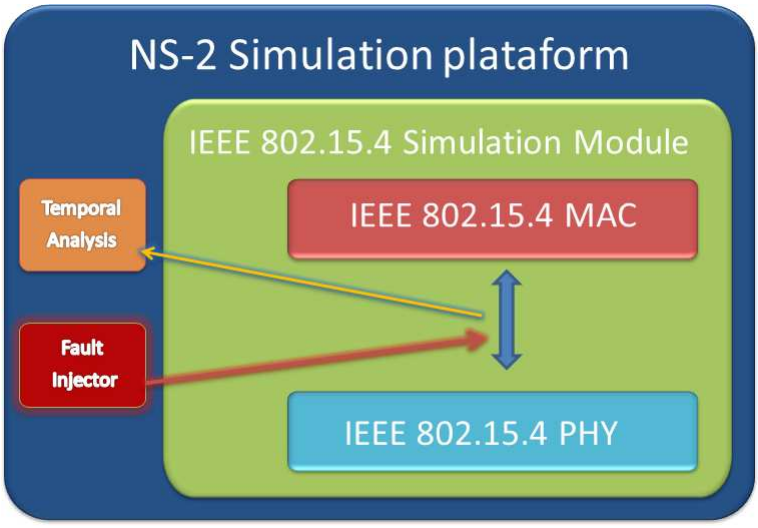

Figure 5.1: New Features in IEEE 802.15.4 module

we needed a tool capable of simulate the inaccessibility scenarios described previously in chapter 3. To simulate some of that scenarios the simulator has to be able to disturb the normal network operation, more specifically, affect the MAC control frames. The NS-2

already provides an error model, however this model cannot affect a specific frame such as MAC control frame. To overcame the current error model limitation, we complement the existing NS-2 components with the integration of new features, fault injector and temporal analysis, in the NS-2 but separated from IEEE 802.15.4 module, as represented in the figure 5.1.

5.2 Injecting faults to simulate accidental errors on the network operation

Our fault injector is capable to use a fault pattern to inject errors during the simulation. The criteria to define the fault pattern is totally configurable, allowing the definition of deterministic or probabilistic fault patterns. A illustration of the fault injection scheme is shown in the figure 5.2.

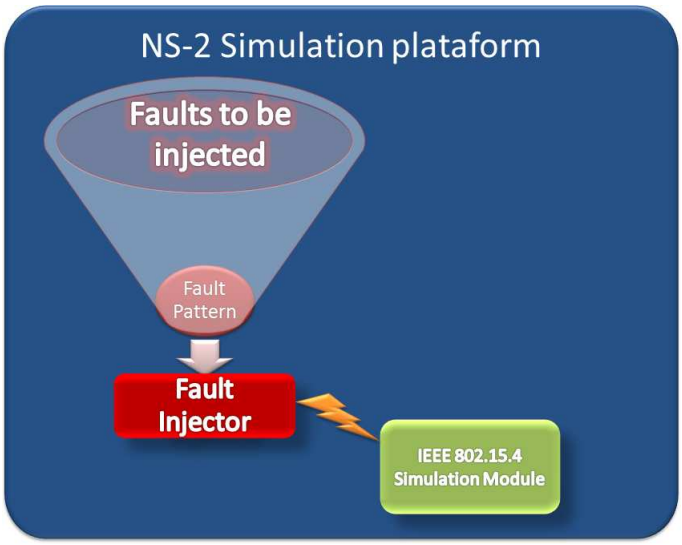

Figure 5.2: Fault Injector scheme

As an example from the faults that can be injected, illustrated in the figure 5.2, a fault pattern can be defined to provoke transmission errors randomly in time (random noise or interference) or be localized in specific time intervals (deterministic noise). On both of these patterns, the fault injector can be customized regarding the type of frame to affect, the rate and the duration of the fault injection.

Patterns with long duration are discouraged for deterministic error models, since such long duration may cause a permanent inaccessibility to the network access if the affected frame is MAC control frame. For example, if we are corrupting a beacon frame injecting deterministic faults successively over a long period we may cause the loss of synchronization by the node and consequently this becoming unable to access the network again.

However this type of pattern is beyond the scope of this work that is to analyse accidental faults where such pattern does not happen.

To perform the random noise or interference is possible to simulate aleatory errors on the network communication, injecting faults between the MAC and the Physical layer (PHY). A random function implemented in the fault injector allows inserting random corruption events in the NS-2 scheduler as described in Algorithm:3. In case of random noise the instant when the corruption occurs is totally aleatory, and is generated through a seed given by argument as described in line 2. A new event is created and the action

Algorithm 3 Fault Injector - A random function

1: Begin.
2: $randomTime = randomGenerator(seed)$;
3: $NewRandomEvent = faultInjector(frameToCorrupt)$;
4: $Scheduler :: instance().schedule(NewRandomEvent, randomTime)$;
5: $CorruptNode.Update()$;
6: End.

associated with it is a frame corruption performed by the fault injector as indicated in line 3. Finally the $NewRandomEvent$ which will perform the corruption is inserted in the NS-2 scheduler and executed at the defined instant as in line 4. An information about the corruption occurred in a specific node is recorded as described in line 5.

The fault injector achieves the frame corruption as described in algorithm:4, accessing the command header of the frame as represented in line 5, and changing a bit in the frame content, implying the drop of these frames in the MAC level of the receiving nodes. When the frame is received if the fault injector is active, we can decide if a specific frame is affected or any frame that a node receives will be corrupted. The parameter $frameToCorrupt$ represented on line 3 is previously defined and if desired all the received frames can be affected defining the $frameToCorrupt$ to a specific value. An information about the corruption occurred in a specific node is recorded as described in line 6. This information is used for a better control of the simulation events. The

Algorithm 4 Fault Injector Mechanism

1: Begin.
2: $MAC.Receive(frame)$;
3: **if** $frame = frameToCorrupt$ **then**
4: **when** selected Fault Pattern **do**
5: $CommandHeader(frame) -> error() = 1$;
6: $CorruptNode.Update()$;
7: **end when**
8: **end if**
9: End.

fault injection may be performed in the coordinator, which implies, depending on the type of frame affected, that the whole network may be inaccessible, in the specific case of affecting a MAC control frame. In case we decide to affect a MAC control frame,

affecting specific network points, the fault injection can be performed for example at non-coordinator nodes tracking the reception of beacon frames. In the specific case, when we perform corruption in a MAC control frame such as the beacon in the coordinator, none of the nodes receives the beacon and therefore the whole network will be inaccessible. Otherwise, when the corruption is performed in the nodes that should receive beacon frames, only the node that has the fault injector component activated, i.e. beacon corruptions occurring, cannot access the medium and becomes inaccessible. The corruption of the frames can be disabled, through the deactivation of the fault injector on the *tcl* script, and the normal behaviour of the network restored at any time.

5.3 Temporal and Energetic Analysis under error conditions

Additionally, to measure the effects of the frame corruption performed by the fault injector in the MAC level, for example the duration of the inaccessibility scenarios, we instrumented the temporal account module. This is responsible to evaluate time events, for instance the periods of inaccessibility.

The duration of the inaccessibility event, for example a beacon loss scenario described in chapter 3. The time is accounted from the instant that the frame is received, and is checked if the beacon is corrupt. If so, the time measurement service component starts a timer to account for the duration of such network inaccessibility period. Because it is a beacon loss scenario the number of lost beacons is also taken into account. The temporal account module are able to consider other time events, such as the reception or loss of other MAC control frames, like the fault injector the temporal account module is able to process generic events. Along with the temporal account module, the energy consumed

Algorithm 5 Record Event Energy, MAC control frame loss scenario

```
1: Begin.
2: MAC.Receive(frame);
3: if frame = FrmCtrl and FrmCtrl is Corrupt then
4:     event.Energy(FrmCtrl, Scenario);
5:     EnergyAccount = CURRENT_ENERGY and CURRENT_TIME
6:     when selected Scenario do
7:         event.Energy.Report();
8:     end when
9: end if
10: End.
```

by the nodes during the simulation events, for example the inaccessibility events, is also recorded by the energetic account module as shown in algorithm:5. For instance, in an inaccessibility event it is verified if the received frame is corrupt, in the line 3 is illustrated the specific case where a MAC control frame is corrupted, the energy spent in the given

scenarios, described in chapter 3, is recorded as well as the time. For the selected scenario as described in the line 6, is possible to generate a report with an energetic analysis of the occurred event.

For both modules, temporal and energetic, a successful re-establishment of the MAC layer communication services indicates the end of a network inaccessibility, and therefore stopping the timer that reveals the duration and energy consumed of its correspondent network inaccessibility event.

Both the temporal and energetic account tool, produce a report regarding the specific event. The report show details about a specific event, or for instance all the inaccessibility events that may occur during the simulation. The log file generated by each execution of the simulation, is used as input to a *gnuplot* script which produces a graphic analysis of the inaccessibility events.

5.4 Design and implementation of the solution

The implementation was made adding new classes to the native NS-2 simulator as represented in figure 5.3. The *faultInjector* class was created, as well as two auxiliary classes *randomCorruption* and *corruptNode*. The first allows the Fault Injector to randomly inject faults in the communication given the simulation time and the desired number of faults. Together with the faultInjector component, allows defining different fault patterns. The *randomCorruption* implements all the inherited methods of the *Handler* class, which is the based class of all network objects and indicate the action to be executed when the event occurs. This allow us to insert the events in the time line of the NS-2 scheduler.

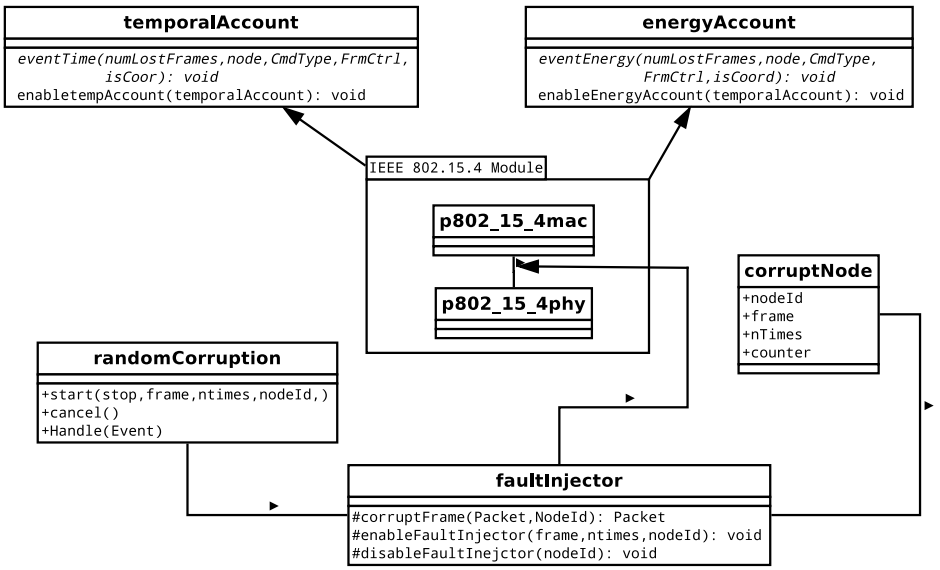

Figure 5.3: Fault Injector Class Model

The second represents a corrupt node, and his attributes allows the fault injector to

control the current execution. Providing information about the current affected node, how much time it has been corrupted and the type of frame that is being affected in the current simulation.

Two other classes were developed to support the analysis of the simulation events, the *temporalAccount* which for example, records the duration of the inaccessibility events and was developed using the internal timer class of the simulator. And the *energyAccount* which provides information about the energy spent during such events and was developed using the current energy model of the simulator.

5.5 Simulating Inaccessibility Scenarios

In order to simulate the network inaccessibility scenarios described in chapter 3, we configure our fault injector component to generate deterministic faults. Starting with the corruption of beacon frames and consequently the loss of beacons. Thus, a know number of beacons is corrupted after the association of all nodes with coordinator, causing the occurrence of the inaccessibility scenarios. The simulation is defined in an OTcl script (Listing:5.1) and is carried out in an one-hop star topology, where all the nodes are within the range of each other.

```
1 Event at 0.0   node_(0) startWnSCoordinator $beaconOrder
    $superFrameOrder";
2 Event at 20.0 node_(1) & node_(2) startDevice"
3 Event at 20.0 node_(1)  enableTemporalAccount $Scenario";
4 Event at 30.0 node_(0) startBeaconTransmission $beaconOrder
    $superFrameOrder"
5 Event at 30.0 node_(1) GTS On"
6 Event at 30.0 node_(1) Start Fault Injection $Beacon $Rounds"
7 Event at $stopTime "stop"
```

Listing 5.1: NS-2 Simulation Script

In the script (Listing:5.1) we define that the first node to start was the coordinator, defining his BO and SO in line 1, then after the WnS is established we start the nodes in line 2. Our temporal account module is enabled on line 3, given the selected scenario. The periodic beacon transmission is initiated at the coordinator on line 4, taking the BO and SO as arguments. At line 5 we enable the GTS transmission for the node(1), which means that hereinafter each time this node have data to transmit will use the GTS mechanism. Finally at line 6 we start our fault injector to, in this example, corrupt beacon frames for a certain number of rounds.

For each addressed scenario we set our fault injector to corrupt a specific frame at a given number of times, on a chosen node. The fault injector can corrupt one of each frame type present in the Table: 5.1 and described in section 2.1.1.

Frame type value	Command frame ID	Standard Reference
0		Beacon
1		Data
2		Ack
3		MAC Control Frame
	01	Association Request
	02	Association Response
	03	Disassociation notification
	04	Data Request
	05	Coordinator conflict notification
	06	Orphan notification
	07	Beacon request
	08	Coordinator realignment
	09	GTS request

Table 5.1: MAC frame types

To achieve the **Single Beacon Frame Loss (SBFL)** scenario we executed the following schedule of Events:

```
1 Event at 30.0 node_(1) Start Fault Injection $Beacon $SBFL"
```

Which means the beacon frame will be corrupted $SBFL$ number of times, corresponding to the current scenario, at the Node(1) after the 30 simulations seconds.

The **Multiple Beacon Frame Loss (MBFL)** happens when we change the number of corrupting rounds on the fault injector depending on the value that $MBFL$ assumes in order to achieve the loss of $nrLost$ beacons. The **synchronization loss** is a special case of the MBFL scenario where after the loss of $nrLost$ beacons the next beacon is also lost.

```
1 Event at 30.0 node_(1) Start Fault Injection $Beacon $MBFL"
```

The **Orphan notification and Coordinator realignment** are achieved when the fault injector corrupts $NOSYNC$ beacon frames, corresponding to the current scenario, and the node lose the synchronization. The Orphan notification is observed on the node and the Coordinator realignment is transmitted by the coordinator on response.

```
1 Event at 30.0 node_(1) Start Fault Injection $Beacon $NOSYNC"
```

So that **Coordinator Conflict Detection** can occur, this event has to be forced on the simulator. Once every time a node becomes a coordinator it assumes its ID as the *networkID*, so a coordinator conflict is impossible because every coordinator assumes a

distinct ID. To force that event we oblige the coordinator to use the same identifier with the following line.

```
1 Event at 0.0 node_(1) Coordinator Conflict 1"
2 Event at 0.0 node_(0) Coordinator Conflict 1"
```

When the **GTS mechanism** is previously activated from the script, and the node has data to transmit, a GTS Request will occur. This request will be send to the coordinator by the node to perform an allocation of a GTS slot for exclusive transmission time.

```
1 Event at 30.0 node_(1) GTS On"
```

5.6 Inaccessibility Results

5.6.1 Simulation Setup

Regarding the Inaccessibility simulation environment, the simulation script (Listing:5.1) was executed in order to achieve the duration of the inaccessibility scenarios. The network was simulated with seven nodes, where one of these nodes, in the center, was the coordinator. All other nodes are in the radio transmission range of the coordinator. Additionally all nodes are in a single broadcast domain, which means that all the nodes are within the range of each other. Our first set of simulation, address the inaccessibility scenarios, a beacon order $BO = SO = 4$ was utilized. The characteristics of the simulation setup scenario are shown in Table 5.2.

Simulation Parameters	
NS-2 Version	2.35 updated with Fault Injector/ Temporal Analysis tool features
Network Topology	Star Topology
Nodes	7
Traffic	Constant Bit Rate (CBR)
Reception range	15m
Carrier Sense range	15m
Packet Size	70 bytes
Transmission/Reception Power	30mW
Beacon	Enabled
Beacon Order	4
Superframe Order	4
Maximum CSMA/CA Attempts	4
Simulation Time	600 seconds

Table 5.2: Simulation Parameters for Inaccessibility Scenarios with BO=SO=4

5.6.2 Simulation Results

After the environment setup, the simulation was performed to obtain the best and worst case duration of the inaccessibility scenarios. Table 5.3 represents a comparison between

Inaccessibility Scenarios Comparison in s			
Scenario	Theoretical Best	Theoretical Worst	Simulated
Single Beacon Frame Loss	0.262	0.262	0.245
Multiple Beacon Frame Loss	0.262	1.045	0.737
Synchronization Loss	1.045	1.045	0.737
Orphan Node	1.099	9.527	8.148
Coordinator Realignment	0.028	0.150	0.017
GTS Request	0.002	0.120	0.001
Coordinator Conflict Detection	0.003	0.126	0.001
Coordinator Conflict Resolution	0.518	8.354	8.052
Association	0.522	8.605	7.717
Re-Association	1.567	9.649	8.270

Table 5.3: Theoretical best and worst case and simulated results for each network inaccessibility scenario with BO=SO=4 and $\mathcal{T}_{BI}$ = 0.240s

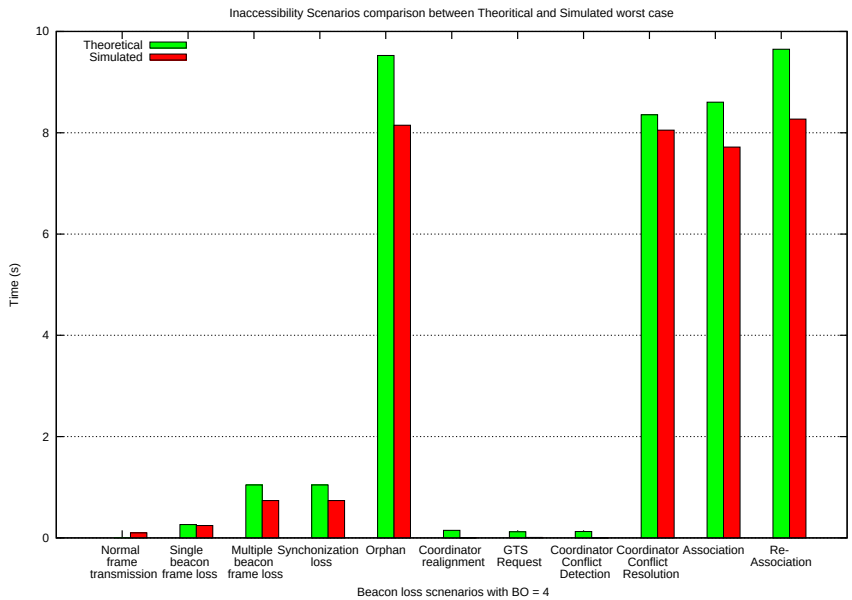

Figure 5.4: Inaccessibility Scenarios comparison between Theoretical and Simulated worst case and BO=SO=4 and $\mathcal{T}_{BI}$ = 0.240s

the simulated obtained values and the theoretical best an worst case obtained by the computation of the formulas presented in chapter 3. It is possible to verify that the theoretical

values present the upper bound of the network inaccessibility scenarios compared with the simulation results.

Figure 5.4 presents the inaccessibility durations values for theoretical and simulated worst case. This results, in comparison with a normal transmission (0.004s) present much higher values. Thus, analysing the figure 5.4 we observe the higher values of the inaccessibility duration events are mainly for the beacon loss related scenarios. The Orphan Node and the Re-Association are by far the events with the most impact on the network. The graphic presented in figure 5.4 show a very high value to meet the requirements of real-time applications. The normalized inaccessibility scenarios with BO=SO=4 com-

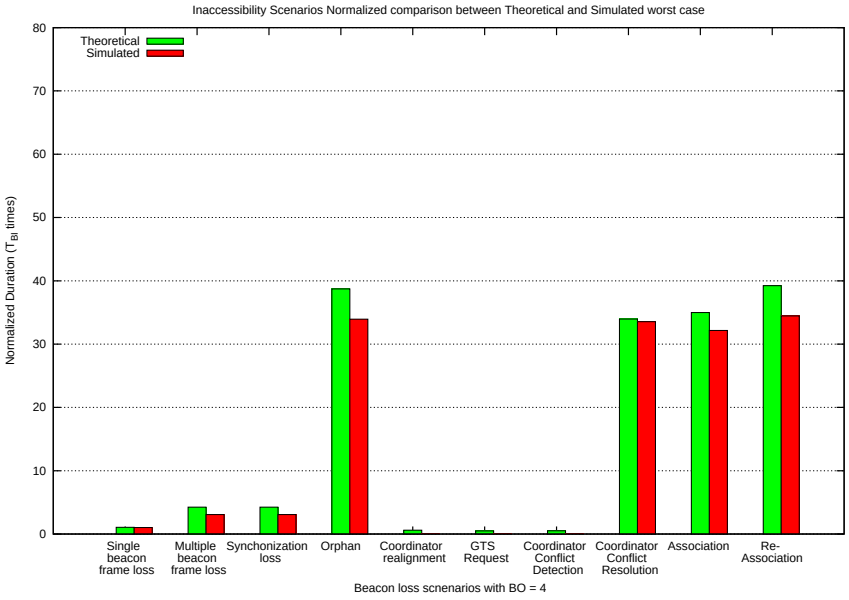

Figure 5.5: Normalized Inaccessibility Scenarios comparison between Theoretical and Simulated worst case and BO=SO=4 and $\mathcal{T}_{BI}$ = 0.240s

parison between theoretical and simulated are presented in figure 5.5 and with more detail in table 5.4.

With the network configuration presented on table 5.2, the simulated worst case period of network inaccessibility is up to ten times higher than the beacon interval. However, it should be noted that the beacon interval is in the order of the seconds, once again, a very high value to meet the requirements of most real-time applications. A beacon order BO=SO=3 is used in figure 5.6 since it's the minimum value for a real-time operation and the characteristics of the simulation setup scenario are shown in Table 5.5. These results clearly show that the periods of inaccessibility are much longer than data frame transmission delays (0.004s) obtained assuming the network is operating normally and therefore inaccessibility has a non negligible impact in network real-time operation. In the figure 5.6 and figure 5.7 we can observe that, with $BO = SO = 3$, the duration on the nodes of the inaccessibility events decrease in comparison with $BO = SO = 4$. Is

Inaccessibility Scenarios Comparison in Periods ($\mathcal{T}_{BI}$)			
Scenario	Theoretical Best	Theoretical Worst	Simulated
Single Beacon Frame Loss	1.065	1.065	1.021
Multiple Beacon Frame Loss	1.065	4.248	3.071
Synchronization Loss	4.248	4.248	3.071
Orphan Node	4.467	38.728	33.950
Coordinator Realignment	0.114	0.610	0.007
GTS Request	0.008	0.488	0.004
Coordinator Conflict Detection	0.012	0.512	0.005
Coordinator Conflict Resolution	2.106	33.959	33.550
Association	2.122	34.980	32.154
Re-Association	6.370	39.224	34.458

Table 5.4: Normalized theoretical best and worst case results for each network inaccessibility scenario with BO=SO=4 and $\mathcal{T}_{BI}$ = 0.240s

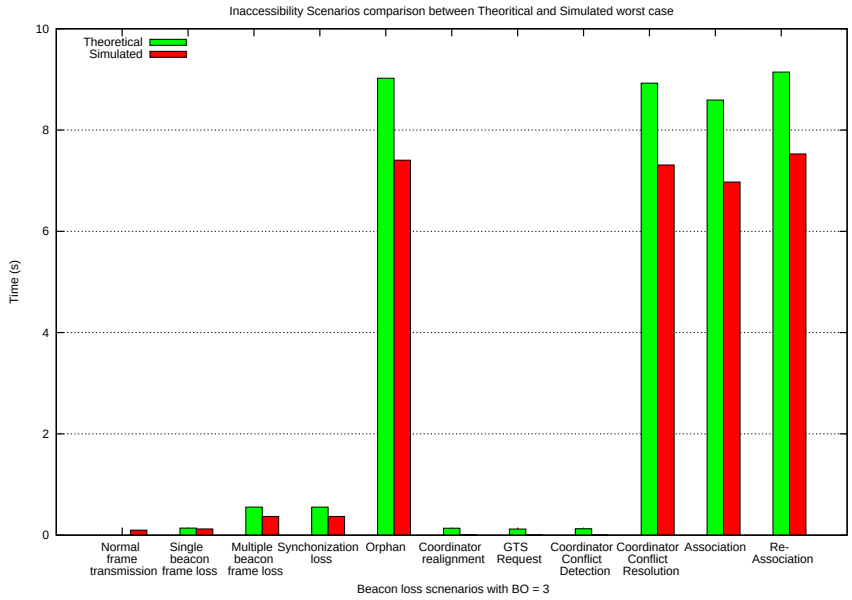

Figure 5.6: Inaccessibility Scenarios comparison between Theoretical and Simulated worst case and BO=SO=3 and $\mathcal{T}_{BI}$ = 0.120s

important to recap that the beacon interval ($\mathcal{T}_{BI}$) is calculated based on the value of BO as described in equation:

$$\mathcal{T}_{BI} = 2^{BO} \times \mathcal{T}_{BSD} \tag{5.1}$$

Hence, the base formula to calculate the $\mathcal{T}_{BI}$ has a exponential component (2^{BO}) where the BO is the exponent. As the duration of beacon based inaccessibility scenarios are multiple of the $\mathcal{T}_{BI}$, and $\mathcal{T}_{BI}$ increases exponentially with the increasing of the BO, these durations increase in the same way.

Simulation Parameters	
NS-2 Version	2.35 updated with Fault Injector/ Temporal Analysis tool features
Network Topology	Star Topology
Nodes	7
Traffic	Constant Bit Rate (CBR)
Reception range	15m
Carrier Sense range	15m
Packet Size	70 bytes
Transmission/Reception Power	30mW
Beacon	Enabled
Beacon Order	3
Superframe Order	3
Maximum CSMA/CA Attempts	4
Simulation Time	600 seconds

Table 5.5: Simulation Parameters for Inaccessibility Scenarios with BO=SO=3

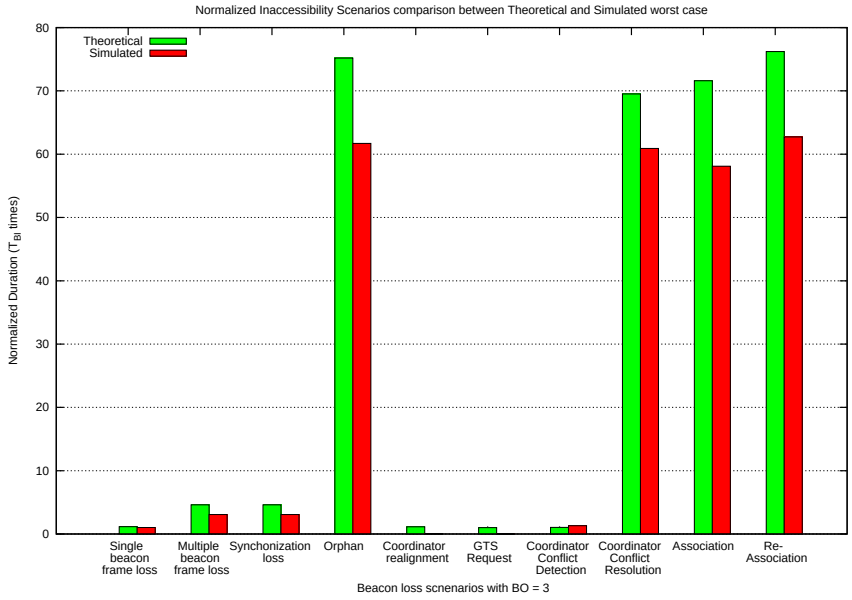

Figure 5.7: Normalized Inaccessibility Scenarios comparison between Theoretical and Simulated worst case and BO=SO=3 and $\mathcal{T}_{BI}$ = 0.120s

If the beacon interval is reduced as represented in figure 5.7 and table 5.7 that presents the normalized inaccessibility scenarios with BO=SO=3, the gap between normal network access times and the periods of network inaccessibility become even higher being the highest values about sixty times the value of the beacon interval. Which implies that the overall system predictability, timeliness and dependability properties may be at risk.

Using the energy model in NS-2, allow us to presents an energy analysis consumption

Inaccessibility Scenarios Comparison in s			
Scenario	Theoretical Best	Theoretical Worst	Simulated
Single Beacon Frame Loss	0.139	0.139	0.122
Multiple Beacon Frame Loss	0.139	0.554	0.368
Synchronization Loss	0.554	0.554	0.368
Orphan Node	0.582	9.023	7.404
Coordinator Realignment	0.015	0.137	0.004
GTS Request	0.002	0.120	0.001
Coordinator Conflict Detection	0.003	0.126	0.001
Coordinator Conflict Resolution	0.505	8.341	7.308
Association	0.509	8.592	6.973
Re-Association	1.062	9.145	7.526

Table 5.6: Theoretical best and worst case and simulated results for each network inaccessibility scenario with BO=SO=3 and $\mathcal{T}_{BI}$ = 0.120s

Inaccessibility Scenarios Comparison in Periods ($\mathcal{T}_{BI}$)			
Scenario	Theoretical Best	Theoretical Worst	Simulated
Single Beacon Frame Loss	1.150	1.160	1.010
Multiple Beacon Frame Loss	1.150	4.600	3.060
Synchronization Loss	4.600	4.600	3.060
Orphan Node	4.850	75.190	61.700
Coordinator Realignment	0.125	1.140	0.032
GTS Request	0.016	1.000	0.010
Coordinator Conflict Detection	0.025	1.050	1.300
Coordinator Conflict Resolution	4.200	69.500	60.900
Association	4.200	71.600	58.100
Re-Association	8.850	76.200	62.710

Table 5.7: Normalized theoretical best and worst case results for each network inaccessibility scenario with BO=SO=3 and $\mathcal{T}_{BI}$ = 0.120s

during the inaccessibility scenarios as illustrated in figure 5.8. The energy analysis is always related to the node experiencing the inaccessibility event. We can observe that the energy consumed by the node greatly increases with the inaccessibility events related with the beacon loss, being the Re-Association the event that most expended energy during the simulation. On the figure 5.9 we can observe that with the lower BO, the duty-cycle, the active period, increase and consequently the activity time of nodes too. Once again we can observe the negative impact of inaccessibility events on the energy consumption of the nodes. The worst case scenarios are related to forced scanning procedures such as

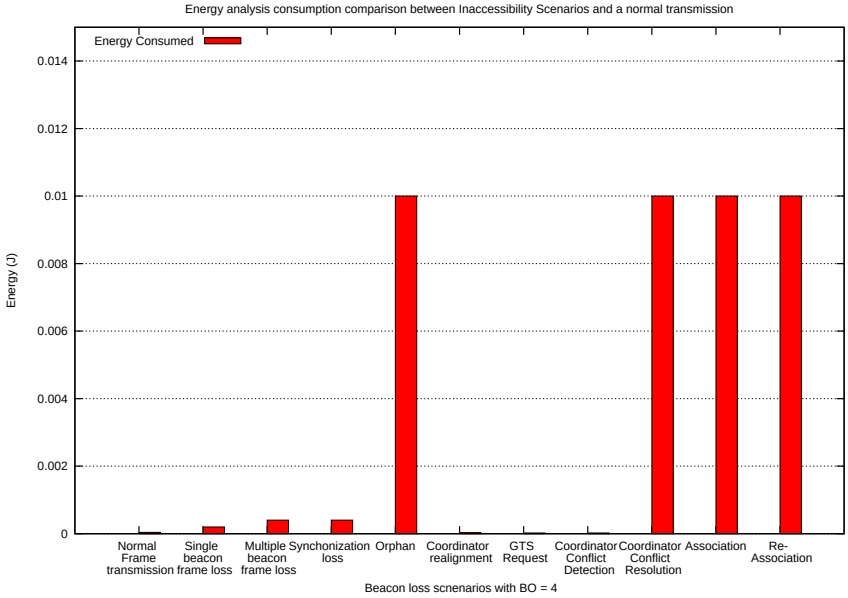

Figure 5.8: Energy analysis consumption of Inaccessibility Scenarios with BO=4

Orphan node, Coordinator conflict resolution, Association and Re-Association. All this scenarios imply more uptime from the node transceivers.

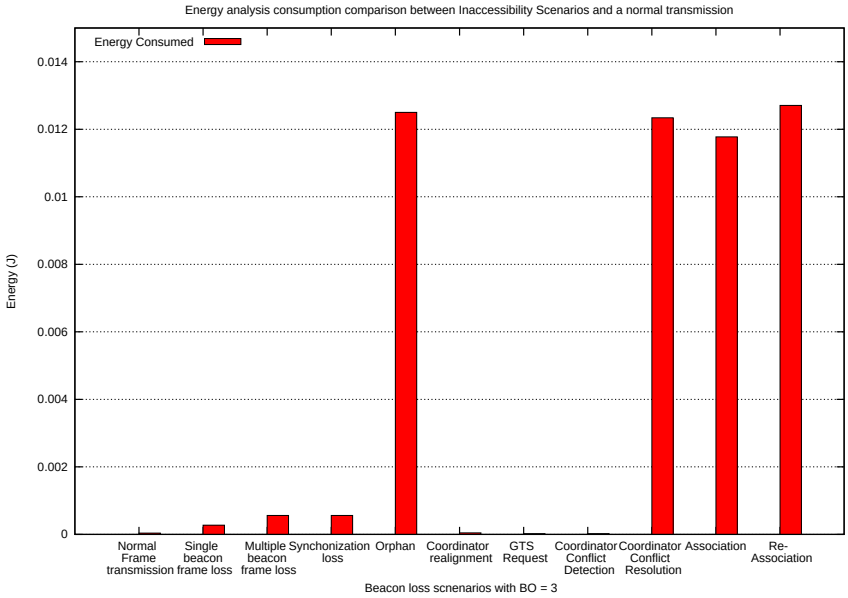

Figure 5.9: Energy analysis consumption of Inaccessibility Scenarios with BO=3

5.7 Summary

This chapter described our simulation results and the validation of a previous theoretical study. We presented, absolute and normalized results regarding the duration of inaccessibility events as well as energetic analyses about the impact of inaccessibility on the node energy consumption.

All the simulated results are lower than the theoretical, this can be explained by deterministic behaviour of the network simulator.

The WSNs are severely limited in terms of power consumption, which makes energy efficiency a very important design requirement. The presented results show that inaccessibility events greatly increase the power consumption on the nodes.

With the potential of the WSNs to support the communication in scenarios with temporal restrictions, this validation is important, allowing the provision of important simulation values about inaccessibility durations and energetic consumptions. This results assist in the characterization of relevant temporal aspects of the communication infrastructure, helping the choice to use of the IEEE 802.15.4 face to the temporal requirements needed by the application executed on top of the communication infrastructure.

Chapter 6

Conclusion

The objective of this thesis was through simulation validate the results obtained in the previous theoretical study about network inaccessibility in IEEE 802.15.4 wireless communications, by providing tools capable of measure network inaccessibility on a simulation environment. In this way significant improvement and modifications in the NS-2 simulator IEEE 802.15.4 module were presented, as well as new module allowing the corruption of specific frames which is not possible with the current error model, and without which we could not perform the simulation and evaluation of all network inaccessibility scenarios. The current IEEE 802.15.4 module implemented in NS-2 is modified and extended to include the use of the GTS mechanisms based on the standard. So, the operations of the GTS allocation, use and deallocation are implemented. The addition of unimplemented MAC operations enhanced the simulation module so that is in accordance to the standard.

Based on NS-2 simulations, we evaluate the performance of various features in the IEEE 802.15.4 MAC. We find that data transmission during the CAP reduces energy cost due to idle listening in the backoff period but increases the collision at higher rate and larger number of sources. While the use of GTS in the CFP can allow dedicated bandwidth to a device to ensure low latency, the device need to track the beacon frames in this mode, which increases the energy cost. The addition of available channels to scan during association revealed an increase of the association time an energy cost, but made the NS-2 more compliant to the standard.

In current simulations, rare features about energy consumption can be specified. If we could get more ways to control the features of energy consumption mechanisms, the simulation will be able to reveal the real situation better. However the presented results show that inaccessibility events greatly increase the power consumption on the nodes and because the energy efficiency is a very important design requirement in WSN this factor cannot be overlooked.

With this simulation, the previous theoretical model[31] was validated providing a fundamental source of information about relevant temporal aspects of the IEEE 802.15.4 beacon-enabled networks. Being aware of the worst case, in network inaccessibility sce-

narios allow us to establish a known bound, so we can do better analysis and definition of a robust timeliness model, in order to achieve an effective support to real-time operation in IEEE 802.15.4 networks.

This greatly assists the IEEE 802.15.4 standard related research. To benefit the research community, our NS-2 implementation of this protocol is publicly available online at: `http://www.karyon-project.eu/wp-content/uploads/2012/10/ns-2-2.35-with-gts.tar.gz`.

Bibliography

[1] G. Anastasi, M. Conti, and M. Di Francesco. A Comprehensive Analysis of the MAC Unreliability Problem in IEEE 802.15.4 Wireless Sensor Networks. *Industrial Informatics, IEEE Transactions*, February 2011.

[2] M. Aoun and P. van der Stok. Overloading an IEEE 802.15.4 Point-to-Point Connection with Real-Time Messages. *31st IEEE Real-Time Systems Symposium*, November 2010.

[3] E. Casilari, J. Cano-García, and G. Campos-Garrido. Modeling of current consumption in 802.15.4/ZigBee sensor motes. *Sensors (Basel, Switzerland)*, June 2010.

[4] C. Chen, B. Yaw Shih, C. Chang, and A. Chen. Enhanced MAC channel selection to improve performance of IEEE 802.15.4. *International Journal of Innovative Computing, Information and Control*, December 2010.

[5] W. Choi and S. Lee. Implementation of the IEEE 802.15.4 module with CFP in NS-2. *Telecommunication Systems*, April 2013.

[6] D. Dondi, A. Bertacchini, S. Scorcioni, L. Larcher, and P. Pavan. Enhancing Safety in Vehicles with Implement or Trailer using an Autonomous Wireless Sensor Network System. *IEEE International Conference on Vehicular Electronics and Safety*, July 2012.

[7] B. Fateh, M. Govindarasu, and A. Layer. Energy-aware Adaptive MAC Protocol for Real-time Sensor Networks. *IEEE International Conference on Communications (ICC)*, June 2011.

[8] H., Hartenstein, and K.P. Laberteaux. A Tutorial Survey on Vehicular Ad Hoc Networks. *IEEE Communications Magazine*, June 2008.

[9] S. Han, X. Zhu, A. Mok, D. Chen, and M. Nixon. Reliable and Real-Time Communication in Industrial Wireless Mesh Networks. *17th IEEE Real-Time and Embedded Technology and Applications Symposium*, April 2011.

[10] L. Hou and N. Bergmann. System Requirements for Industrial Wireless Sensor Networks . *IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, September 2010.

[11] IEEE 802.15.4. Part 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (WPANs) - IEEE standard 802.15.4. IEEE P802.15 Working Group, 2011.

[12] M. Imran, A. Said, and H. Hasbullah. A survey of simulators, emulators and testbeds for wireless sensor networks. *International Symposium on Information Technology*, June 2010.

[13] T. Issariyakul and E. Hossain. *Introduction to Network Simulator NS2*. Springer US, 2009.

[14] J. Rufino J. L. R. Souza, S. Singh. Characterizing inaccessibility in ieee 802.15.4 through theoretical models and simulation tools. Technical report, AIR-II Technical Report RT-10-08, September 2010.

[15] H. Karl and A. Willig. *Protocols and Architectures for Wireless Sensor Netwoks*. John Wiley Sons, 2007.

[16] M. Korkalainen, M. Sallinen, N. Kärkkäinen, and P. Tukeva. Survey of Wireless Sensor Networks Simulation Tools for Demanding Applications. *Fifth International Conference on Networking and Services*, 2009.

[17] A. Koubaa, M. Alves, B. Nefzi, and Y. Song. Improving the IEEE 802.15.4 Slotted CSMA/CA MAC for Time-Critical Events in Wireless Sensor Networks. Technical report, Polytechnic Institute of Porto, July 2006.

[18] Philip Levis and Nelson Lee. TOSSIM : A Simulator for TinyOS Networks. 2003.

[19] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. Wireless sensor networks for habitat monitoring. *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications - WSNA '02*, 2002.

[20] NS-2. The network simulator version 2 contributed code. `http://nsnam.isi.edu/nsnam/index.php/Contributed_Code/`, 2013. [Online; accessed 26-September-2013].

[21] NS-2. Ns-2: The network simulator version 2. `http://www.isi.edu/nsnam/ns/`, 2013. [Online; accessed 26-September-2013].

[22] NS-3. The network simulator version 3. `http://www.nsnam.org/`, 2013. [Online; accessed 26-September-2013].

[23] Omnet++. Discrete event simulation system. `http://www.omnetpp.org/`, 2013. [Online; accessed 26-September-2013].

[24] OPNET. Opnet:application and network performance tool. `http://www.opnet.com`, 2013. [Online; accessed 26-September-2013].

[25] M. Park, K. Kim, and C. Lee. Holistic Optimization of Real-Time IEEE 802.15.4/ZigBee Networks. *IEEE International Conference on Advanced Information Networking and Applications*, 2011.

[26] Prowler. Probabilistic wireless network simulator. `http://www.isis.vanderbilt.edu/Projects/nest/prowler/`, 2013. [Online; accessed 26-September-2013].

[27] J. Åkerberg, M. Gidlund, and M. Björkman. Future research challenges in wireless sensor and actuator networks targeting industrial automation. In *9th IEEE International Conference on Industrial Informatics (INDIN)*, 2011.

[28] L. Rodrigues and P. Verissimo. AMp : A Highly Parallel Atomic Multicast Protocol. *ACM SIGCOM'89 Symposium*, September 1989.

[29] X. Shuai and Z. Zhang. Research of Real-Time Wireless Networks Control System MAC Protocol. *Journal of Networks*, Apr 2010.

[30] H. Song, Z. Xiuming, A.K. Mok, C. Deji, and M. Nixon. Reliable and real-time communication in industrial wireless mesh networks. In *17th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2011.

[31] J. Souza and J. Rufino. Characterization of inaccessibility in wireless networks - a case study on IEEE 802.15.4 standard. In *IFIP 3th International Embedded System Simposium IESS*, September 2009.

[32] J. Souza and J. Rufino. An approach to enhance the timeliness of wireless communications. In *5th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM)*, Lisbon, 2011.

[33] J. Souza, J. Rufino, and A. Guerreiro. Characterizing Inaccessibility in IEEE 802.15.4 Through Theoretical Models and Simulation Tools. *In 4th Symposium on Informatics (INForum)*, September 2012.

[34] T. Stone, R. Alena, J. Baldwin, and P. Wilson. A viable COTS based wireless architecture for spacecraft avionics. In *IEEE Aerospace Conference*, 2012.

[35] Z. Teng and K. Kim. A Survey on Real-Time MAC Protocols in Wireless Sensor Networks. *Communications and Network*, 2010.

[36] Z. Ting, H. Sharif, M. Hempel, P. Mahasukhon, W. Wei, and M. Tao. A novel adaptive distributed cooperative relaying MAC protocol for vehicular networks. *IEEE Journal on Sel. Areas in Communications.*, 2011.

[37] P. Veríssimo and J. Alves Marques. Reliable broadcast for fault-tolerance on local computer networks. In *Ninth Symposium on Reliable Distributed Systems*. IEEE, 1990.

[38] P. Veríssimo and L. Rodrigues. *Distributed Systems for System Architects*. Kluwer Academic Publishers, 2001.

[39] P. Veríssimo, J. Rufino, and L. Rodrigues. Enforcing real-time behaviour on LAN-based protocols. *10th IFAC Workshop on Distributed Computer Control Systems*, September 1991.

[40] S. Xiao-Ying and Z. Zhen-Chao. Research of real-time wireless networks control system MAC protocol. *Journal of Networks*, 2010.

[41] J. Zheng and M. J. Lee. *A Comprehensive Performance Study of IEEE 802.15.4*, chapter 4, pages 218–237. IEEE Press, Wiley Interscience, June 2006.

[42] X. Zhu, S. Han, P. Huang, A. Mok, and D. Chen. MBStar: A Real-time Communication Protocol for Wireless Body Area Networks. *23rd Euromicro Conference on Real-Time Systems*, 2011.