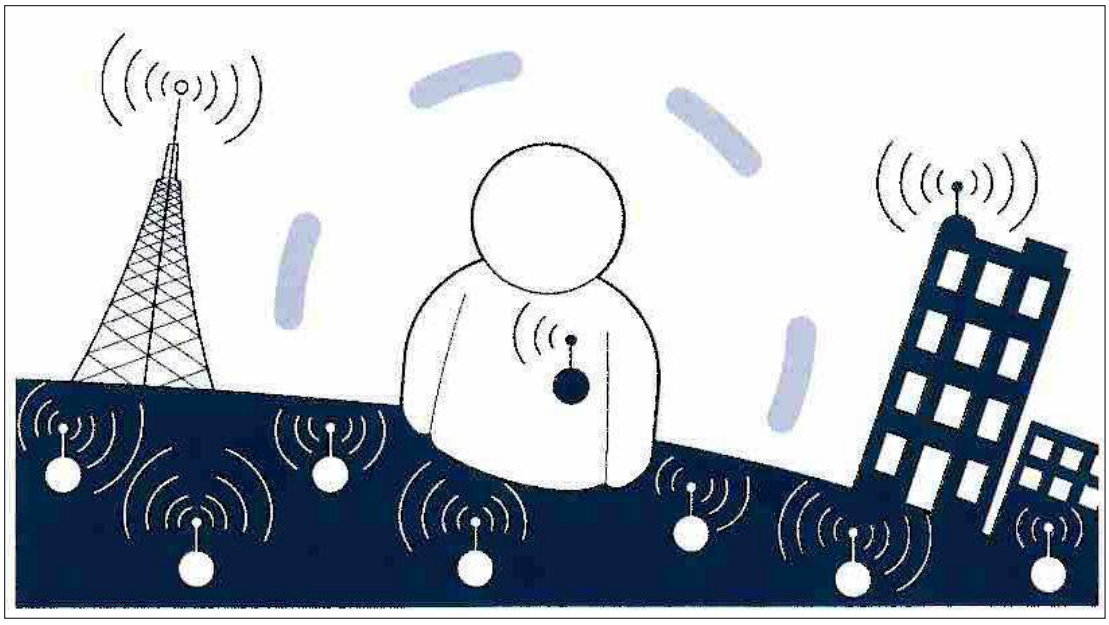
Contribution to quality of service in wireless sensor networks

Marion Souil

Par **Marion SOUIL**

Contribution à la qualité de service dans les réseaux de capteurs sans fil

Thèse présentée
pour l'obtention du grade
de Docteur de l'UTC



Soutenue le 09 octobre 2013
Spécialité : Technologies de l'Information et des Systèmes

D2107

Qualité de Service dans les Réseaux de Capteurs Sans Fil

Quality of Service in Wireless Sensor Networks

Thèse présentée le 9 octobre 2013 pour l'obtention du grade de Docteur de l'UTC

Champ disciplinaire : Technologies de l'Information et des Systèmes

Auteur :
Marion SOUIL

Directeur de thèse :
Prof. Abdelmadjid BOUABDALLAH

Président du jury :
Prof. Bertrand DUCOURTHIAL

Rapporteurs :
Prof. Marcelo DIAS DE AMORIM
Prof. Pascal LORENZ

Examinateurs :
Prof. Ken CHEN
Prof. Yacine CHALLAL

Remerciements

Je tiens tout d'abord à remercier les membres du jury pour le temps qu'ils ont consacré à l'évaluation de ma thèse et tout particulièrement les rapporteurs MM. Pascal LORENZ et Marcelo DIAS DE AMORIM pour leur lecture attentive de ce manuscrit et leurs précieuses remarques. Je remercie les examinateurs MM. Ken CHEN, Yacine CHALLAL et Bertrand DUCOURTHIAL qui m'ont fait l'honneur de participer à ce jury de thèse et pour leurs questions pertinentes qui ont apporté un éclairage nouveau sur mes travaux.

Je remercie chaleureusement l'ensemble du personnel du laboratoire HeuDiaSyC que j'ai eu beaucoup de plaisir à côtoyer pendant ces trois années. J'exprime toute ma gratitude à M. Ali CHARARA, directeur du laboratoire, et M. Aziz MOUKRIM, responsable de l'activité "Réseaux et Optimisation", pour leur accompagnement, leur soutien et leurs encouragements. Je remercie également le personnel administratif et en particulier Céline, Bérengère, Nathalie, Magali et Sabine pour leur gestion efficace des soucis bureaucratiques et surtout pour leur gentillesse.

J'adresse mes sincères remerciements et toute mon amitié à tous les doctorants qui ont évolué à mes côtés, Abdelkrim, Walid, Ahmed, Tifenn, Corentin, Ernesto et bien d'autres encore, dont le soutien moral quotidien m'a permis de surmonter les périodes de doute.

Merci à mes proches, à ma famille et à mes amis, en particulier à Stéphane et à mon mari Damien, qui, en ayant cru en ma réussite et en exprimant continuellement leur confiance en moi, m'ont permis d'avancer et d'aller au bout de ce travail.

Enfin, je remercie infiniment mon directeur de thèse M. Abdelmadjid BOUABDALLAH sans qui cette thèse n'aurait sans doute pas eu lieu. Depuis mon intention de débuter une thèse jusqu'à la soutenance, ainsi que dans l'élaboration de mon projet de carrière, il a su m'épauler, me soutenir, m'encourager, me motiver et m'apporter son expérience et ses conseils. Sa disponibilité, sa confiance et son amitié ont été la clé de ma réussite.

"Learn from yesterday, live for today, hope for tomorrow. The important thing is not to stop questioning."

Albert Einstein

Abstract

The availability of small, low-cost, battery operated devices capable of sensing, performing simple processing and transmitting data via wireless communications have the potential to revolutionize traditional monitoring applications. Wireless networks composed of autonomous sensor nodes enable ubiquitous monitoring tasks from environmental control of office buildings to the detection of forest fires. Recently, new applications for wireless sensor networks such as healthcare and multimedia applications have emerged. These applications often have heterogeneous sensing capabilities and require that the network supports different types of QoS-constrained traffic at variable rates. However, designing efficient protocols that provide an appropriate level of performance to these applications while coping with the limited resources of sensor networks is a challenging task.

In this thesis, we focus on QoS provisioning at the MAC layer. Since this layer is responsible for the organization of channel access, it determines to a large extent the overall performance of the network. We start by studying the specific requirements of demanding and heterogeneous applications, then we discuss related work of the literature. Given the inadequacy of existing solutions in the presence of important traffic loads, we propose AMPH, an adaptive MAC protocol with QoS support for heterogeneous wireless sensor networks. Our solution is a hybrid channel access method based on time division where all nodes may contend to access the channel at each time slot using a new contention mechanism which favors high priority traffic. Through these efficient techniques, AMPH achieves high channel utilization under variable traffic loads and provides low latency to real-time traffic. We verify the efficiency of AMPH through simulation experiments and a mathematical analysis.

Keywords : Wireless sensor networks, quality of service, medium access control, multimedia.

Résumé en français

L'apparition récente de petits capteurs peu couteux fonctionnant sur batteries, capables de traiter les données acquises et de les transmettre par ondes radio ont le potentiel de révolutionner les applications de surveillance traditionnelles. Les réseaux sans fils composés de nœuds capteurs autonomes proches de la cible à surveiller permettent des tâches de surveillance précises allant du contrôle de la température dans des bâtiments jusquà la détection de feux de forêt. Récemment, de nouvelles applications de réseaux de capteurs sans fil telles que des applications multimédia ou dans le domaine de la santé ont émergé. Les réseaux sous-jacents déployés pour ces applications sont souvent composés de nœuds hétérogènes comportant différents capteurs et doivent fournir un niveau de service conforme aux exigences des différents types de trafic en s'adaptant à la charge variable. Cependant, concevoir des protocoles efficaces adaptés à ces applications tout en s'accommodant des ressources limitées des réseaux de capteurs est une tâche difficile.

Dans cette thèse, nous nous focalisons sur le support de la qualité de service au niveau de la couche MAC, car cette couche conditionne et détermine largement les performances du réseau étant donné qu'elle est responsable de l'organisation de l'accès au canal. Dans un premier temps, nous étudions les contraintes spécifiques des applications ayant des exigences fortes ainsi que des applications hétérogènes et nous examinons les travaux proposés dans la littérature. Etant donnée l'inadéquation des solutions existantes en présence dun trafic important, nous proposons AMPH, un protocole MAC adaptatif avec qualité de service pour les réseaux de capteurs sans fil hétérogènes. Notre solution consiste en une méthode daccès au canal hybride basée sur le multiplexage temporel, dans laquelle tous les nœuds peuvent accéder au canal à chaque division de temps en utilisant un nouveau mécanisme de compétition qui favorise le trafic prioritaire. Grâce à ces techniques, AMPH utilise efficacement le canal quelque soit la charge de trafic et assure une latence faible au trafic temps réel. Nous vérifions les performances de AMPH à l'aide de simulations et d'un modèle mathématique.

Mots-clés : Réseaux de capteurs sans fil, qualité de service, medium access control, multimédia.

List of publications

International journals

M. Souil, A. Bouabdallah, A.E. Kamal. *Efficient QoS Provisioning at the MAC Layer in Heterogeneous Wireless Sensor Networks*. Submitted to Computer Communications.

A. Hadjidj, M. Souil, A. Bouabdallah, Y. Challal, H. Owen. *Wireless Sensor Networks for Rehabilitation Application: Challenges and Opportunities*. Journal of Network and Computer Applications (JNCA), Elsevier, vol. 36, No. 01, pp. 1-15, 2013.

International conferences

M. Souil, T. Rault, A. Bouabdallah. *A New Adaptive MAC Protocol with QoS support for Heterogeneous Wireless Sensor Networks*. 17th IEEE Symposium on Computers and Communications, July 2012, Cappadocia, Turkey.

M. Souil, A. Bouabdallah. *On QoS Provisioning in Context-Aware Wireless Sensor Networks for Healthcare*. 20th International Conference on Computer Communications and Networks, August 2011, Lahaina, HI, USA.

Table of contents

List of figures

List of tables

Abbreviations

AC	**A**cess **C**ategory
ACK	**ACK**nowledgment
AI	**A**rtificial **I**ntelligence
BAN	**B**ody **A**rea **N**etwork
CCA	**C**lear **C**hannel **A**ssessment
CLI	**C**ommand-**L**ine **I**nterface
CMOS	**C**omplementary **M**etal **O**xide **S**emi-conductor
CSMA/CA	**C**arrier **S**ense **M**ultiple **A**ccess with **C**ollision **A**voidance
CTS	**C**lear **T**o **S**end
CW	**C**ontention **W**indow
ECN	**E**xplicit **C**ongestion **N**otification
ECG	**E**lectro**c**ardio**g**raphy
EDCA	**E**nhanced **D**istributed **C**hannel **A**ccess
EDF	**E**arliest **D**eadline **F**irst
EMG	**E**lectro**m**yo**g**raphy
FCFS	**F**isrt **C**ome **F**isrt **S**erved
FDMA	**F**requency **D**ivision **M**ultiple **A**ccess
FIFO	**F**isrt **I**n **F**isrt **O**ut
GUI	**G**raphical **U**ser **I**nterface
IDE	**I**ntegrated **D**evelopment **E**nvironment
IP	**I**nternet **P**rotocol
LLC	**L**ogical **L**ink **C**ontrol
MAC	**M**edia **A**ccess **C**ontrol
QoS	**Q**uality **o**f **S**ervice

RED	**R**andom **E**arly **D**rop
RFID	**R**adio **F**requency **ID**entification
RSSI	**R**eceived **S**ignal **S**trength **I**ndication
RTS	**R**equest **T**o **S**end
SP	**S**trict **P**riority
TCP	**T**ransport **C**ontrol **P**rotocol
TDMA	**T**ime **D**ivision **M**ultiple **A**ccess
TXOP	**T**ransmit Opportunity
UDP	**U**ser **D**atagram **P**rotocol
VoIP	**V**oice **o**ver **IP**
VoWLAN	textbfVoice **o**ver **W**ireless **LAN**
WFQ	**W**eighted **F**air **Q**ueuing
WLAN	**W**ireless **L**ocal **A**rea **N**etworks
WMSN	**W**ireless **M**ultimedia **S**ensor **N**etworks
WPAN	**W**ireless **P**ersonal **A**rea **N**etworks
WSAN	**W**ireless **S**ensor and **A**ctor **N**etworks
WSN	**W**ireless **S**ensor **N**etworks

General Introduction

Technology continues to get smaller and faster, and we increasingly find new ways to integrate it into our lives. The advances in sensing and computing capabilities of communicating entities bring new trends and new digital paradigms such as ubiquitous, pervasive mobile computing through smart communicating objects interacting with our environment. In particular, the availability of small, low-cost, battery operated devices capable of sensing, performing simple processing and transmitting data via wireless communications have the potential to revolutionize traditional monitoring applications. The ubiquitous integration of intelligent computing devices forming distributed wireless sensor networks enable a whole new class of autonomous control applications and services from environmental control of office buildings to the detection of forest fires. However, because of the limited resources and singularities of wireless sensor networks, traditional protocols of the TCP/IP stack may not be well suited. In addition, given the diversity of envisioned applications, it is unlikely that there will be a one size fits all communication architecture for all the possible applications. Since each application has specific characteristics, to each of them corresponds a specific network design and specific protocols with specific technological issues.

Until recently, the majority of research work on wireless sensor networks focused on single-purpose applications and aimed to reduce energy consumption in order to maximize the network lifetime. Since these low data rates applications often are delay and loss tolerant, they do not necessitate quality of service support and allow duty cycling. Lately, the development of low-cost hardware such as CMOS cameras, microphones and health sensors has fostered the development of new applications for wireless sensor networks such as healthcare and multimedia applications. These applications often have heterogeneous sensing capabilities and require that the network supports different types of QoS-constrained traffic at variable rates. Providing an appropriate level of service for this new category of applications while coping with the limited resources of wireless sensor networks is a challenging task. Even though efforts have been

made to provide QoS-aware routing and MAC protocols, existing solutions often assume low data rates and address only one specific issue (e.g., latency or reliability improvement), therefore their performance is inadequate in the presence of high rate heterogeneous traffic with multiple QoS requirements.

In this thesis, we focus on heterogeneous wireless sensor networks applications with high QoS requirements such as healthcare and multimedia applications. We are interested in the design of efficient QoS-aware communication protocols in order to provide an appropriate level of service to these applications.

Contributions

In this manuscript, we report our work on quality of service support for wireless sensor networks applications.We focus on QoS provisioning at the MAC layer, since this layer determines to a large extent the overall performance of the network, given that it is responsible for the organization of channel access. We also address heterogeneity support in order to enable WSN applications with various sensing capabilities and traffic types with different QoS requirements.

As a preliminary work, we studied the specific QoS requirements of demanding and heterogeneous WSN applications. In particular, we analyzed the characteristics of healthcare applications and we assessed the state of the art on QoS provisioning for these critical applications. This study highlighted the need for adaptive QoS protocols and frameworks whose design raises numerous challenges.

In order to fill the gap in the literature, we proposed AMPH – an Adaptive MAC Protocol with quality of service support for Heterogeneous wireless sensor networks. AMPH adopts an hybrid behavior which combines the strengths of contention-based and contention-free medium access techniques to achieve high channel utilization. It supports two distinct classes of traffic for real-time and best-effort data flows and gives high priority to real-time traffic through an efficient contention mechanism which relies on random backoffs. This mechanism may easily be adapted for a larger number of traffic classes.AMPH uses no control messages such as RTS/CTS and acknowledgments in order to minimize the MAC latency and to reduce energy consumption. Despite the absence of control messages, in one hop networks, our solution achieves high data delivery ratio without the overhead of complex reliability techniques. However, in multihop scenarios, AMPH suffers from the hidden terminal problem. In order to extend the scope of

our protocol, AMPH v2 implements an enhanced adaptive contention mechanism with collision avoidance that allocates higher bandwidth to nodes having more traffic.

We implemented AMPH in the network simulator OMNeT++ and we assessed its efficiency through extensive simulation experiments. Moreover, in order to further evaluate the performance of AMPH, we proposed a mathematical model of our protocol. The model allows the evaluation of the MAC latency for real-time and best-effort under custom traffic loads. In addition, the model estimates the data delivery ratio while assuming ideal channel conditions. We also implemented AMPH on a real sensor platform to demonstrate the feasibility of our solution.

Manuscript Organization

In the remainder of this manuscript, the presentation of our contributions is organized as follows.

The first chapter of this thesis is dedicated to the general presentation of wireless sensor networks. In this chapter, we present the basics of wireless sensor networks design, communication architecture and protocols, then we expose sensor networks challenges. The aim of this chapter is to help non-specialists in the field to get an overview of wireless sensor networks and their main research issues, and to pave the way for the following deeper study of specific challenges.

In Chapter 2, we define the concept of quality of service, then we study in detail QoS support in traditional data networks and well-known QoS mechanisms. Afterwards, we provide an overview of QoS provisioning approaches in wireless sensor networks, then we assess the state of the art of QoS support for demanding WSN applications, in particular at the MAC layer. Finally, we point out open issues and remaining challenges in QoS provisioning in WSN.

In Chapter 3, we present the motivation and the design of AMPH, an adaptive hybrid MAC protocol with QoS support for heterogeneous WSN. First, we explain the design goals of our solution, then we introduce the basic principles of AMPH, and finally, we provide a detailed overview of the operation and the features of the protocol in order to highlight its benefits.

Chapter 4 reports on the performance evaluation of AMPH through simulation experiments. In this chapter, we provide the description of our approach for implementing the simulations as well as detailed explanations on simulation scenarios and parameters. We analyze the performance of our solution in terms of channel utilization, latency, and data delivery ratio, and we discuss the results by comparing them to those of a well-known contention-based QoS-aware MAC protocol

for wireless multimedia sensor networks. We study favorable conditions for the use of AMPH, along with its limitations. We then highlight areas for improvement of our solution.

In Chapter 5, we perform an analytical study of AMPH. We propose a mathematical model of our solution in order to further analyze its efficiency.

In Chapter 6, we focus on providing multihop support to AMPH. We propose a new contention mechanism that solves the hidden terminal problem and we provide a succinct analysis of the performance of this new algorithm.

Chapter 7 provides an overview of our approach to implement our solution on a real sensor platform and early results of the experimental performance evaluation of AMPH.

Finally, we conclude this thesis by providing a summary of our contributions and discussing prospects and opportunities for the future.

Chapter 1

Wireless Sensor Networks Basics

In this chapter, we introduce the concept of wireless sensor networks (WSN). We present the main fields of application, then we provide an overview of sensor networks design, architecture and protocols. Finally, we review research challenges for WSN.

1.1 Overview

A new class of networks has appeared in the last few years: the wireless sensor networks. These networks consist of individual sensor nodes deployed in a given area that cooperatively collect and carry data to a main entity in order to monitor physical or environmental conditions. The main entity, also denoted as base station or sink, can be connected to an infrastructure or to the Internet through a gateway. The system can also operate without the need for an existing infrastructure, thus the network operates autonomously. The user can periodically collect the gathered data through a direct connection to the sink using a mobile device, such as a laptop or a smartphone.

Sensor nodes are composed of sensing, data storage, data processing, and communicating components powered with batteries. They may be equipped with additional elements such as a localization system (GPS), power harvesting components, etc. The cost of nodes has to be kept low to minimize the overall cost of the network so that WSN solutions are cheaper than traditional networks. A good trade-off must be found between the amount of features and the cost.

WSN have the potential to revolutionize traditional monitoring tasks. Indeed, the availability of such low-cost sensor nodes enables ubiquitous unattended monitoring in areas difficult to

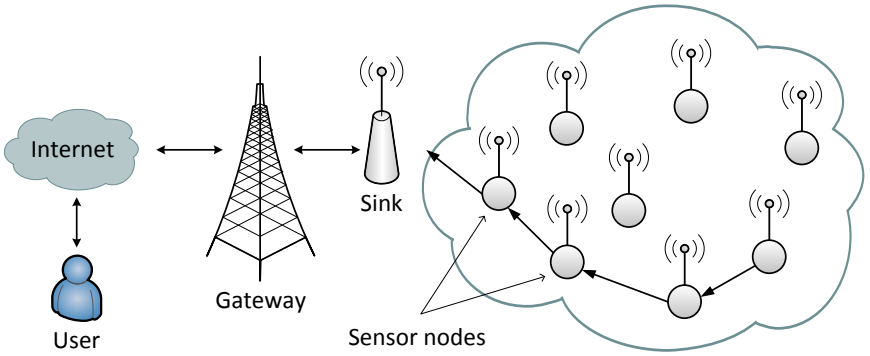


FIGURE 1.1: Architecture of a simple WSN

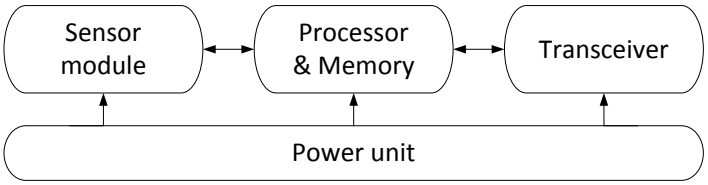


FIGURE 1.2: Architecture of a sensor node

access. Moreover, these networks often include actuators thus allowing the system to interact with the environment leading to pervasive control. However, realizing such wireless sensor networks is a challenging research and engineering problem because of the diversity of envisioned applications and the limited resources of the sensor nodes.

1.2 Application examples

Wireless sensor networks will facilitate many existing applications and enable a lot of new promising ones [1]. As the application field is very wide, this technology has the capability to transform people's lives all over the world. Several different applications can be constructed with different types of nodes and sensing faculties. For many physical parameters, appropriate sensor technology exists that can be integrated in a tiny node. Some popular ones are temperature, humidity, and light sensors, but even more sophisticated sensing capabilities are conceivable, e.g., acoustic, vibration, magnetic sensors, infrared light sensors, cameras, accelerometers, gyroscopes, and medical sensors (heart rate and blood pressure sensors). The field of applications of WSN is only limited by the imagination. In the following section, we highlight some common application scenarios.

1.2.1 Environment

Disaster management applications Disaster monitoring applications such as wildfire detection are typical applications of WSN. As the environment is harsh, sensor nodes can easily be deployed from an airplane over the hazardous area, then the nodes individually detect their location, collect environment readings such as temperature and humidity, and transmit them to a base station in a safe area which computes the global heat map. It is likely that in such scenarios, most sensors nodes are not in the range of the sink. The network must therefore self-organize and construct multihop paths in order to route the data of all nodes to the sink.

Agriculture applications Agriculture can considerably benefit from WSN. Combined air and ground sensors can monitor temperature, humidity, and light information in order to optimize the irrigation and save water but also to prevent plant disease. When using long-range sensor nodes (transmission at low data rates with high transmit power or high gain antennas), only few sensors are needed to cover wide areas (about one sensor per 100m x 100m area).

Biodiversity monitoring applications WSN can be used to observe animal species difficult to approach by humans. Areas of interest are covered with sensor nodes equipped with basic sensors, motion sensors, and cameras. The network is left unattended, sensor nodes autonomously gather data and pictures that are periodically collected on a mobile device (laptop, smartphone). Sensors can also be placed on animals. Such networks are able to detect and track animals and take pictures allows scientists to observe wildlife, monitor and protect endangered species.

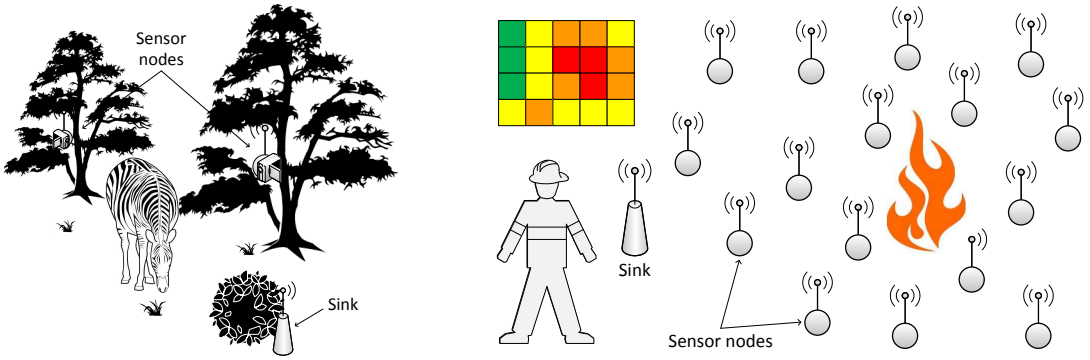


FIGURE 1.3: Illustration of wildlife observation and wildfire detection applications

Intelligent buildings Infrastructures such as buildings and bridges deteriorate over time and require maintenance. However, it is not easy to detect problems. WSN can perform real-time structural health monitoring by detecting temperature changes, steel distortion, earth tremors, etc.

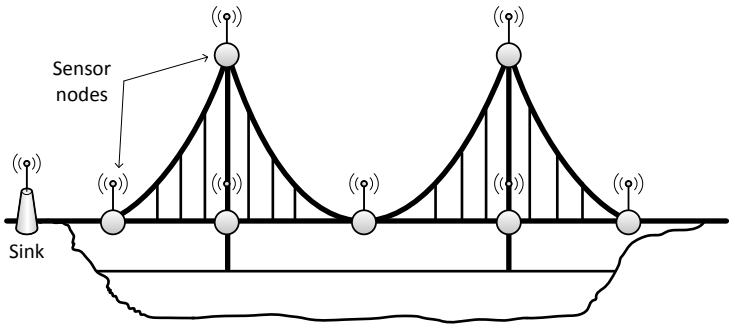


FIGURE 1.4: WSN structural health monitoring application

1.2.2 Industry

Applications for logistics WSN can facilitate tracking of various objects in warehouses and stores. It is possible to equip goods with sensor nodes which can detect checkpoints, store tracking information and can be used to locate items.

Applications for machine surveillance and preventive maintenance The main advantage of WSN in machine surveillance is the cable-free operation. The sensor nodes can be fixed in difficult to reach locations, hazardous or restricted areas where it would be difficult for humans to control the machinery. They can detect corrosion or vibration patterns that indicate the need for maintenance and transmit the collected data and alarms to a wireless base station. As in many WSN application scenarios, wired power may not always be available to supply the nodes.

1.2.3 Military applications

Battlefield surveillance applications A WSN deployed over a strategic area enables the surveillance of every ally or enemy unit as well as site analysis (sensors can detect the presence of nuclear, biological, and chemical agents). WSN applications for battlefield surveillance can thus save the lives of many soldiers

Border control applications It is possible to form a security perimeter with sensor nodes in order to keep intruders out of a certain area. WSN applications for border control can be used to alert patrols or may include robots to track and surround the target.

1.2.4 Health care

Mobile patient monitoring applications Wireless sensors such as blood pressure and heart rate sensors allow continuous and unobtrusive monitoring of physiological parameters. Such sensors, placed on the patient's body with a base station, form a network able to deliver high quality care for patients while allowing autonomy and mobility, for example, helping patients to manage their diabetes and monitoring rehabilitation exercises while removing the burden of wires. Wireless sensor network applications for health care can considerably improve the quality of life of chronically ill, elderly or people with disabilities. WSN in hospitals can also be employed as a patient and doctor tracking system, raising alarms when necessary (doctors or nurses are alerted when the system detects the deterioration of a patient's condition for example).

Telemedicine applications Similar WSN as those employed in hospitals can be used in home environments. These systems allow patients to be treated at home instead of the hospital. Their vital signs are continuously monitored and transmitted to medical staff which can react to emergency situations.

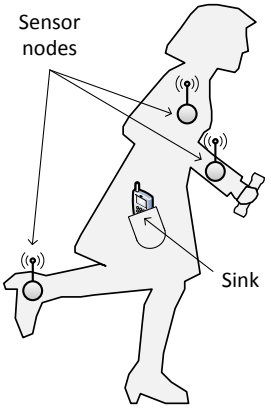


FIGURE 1.5: WSN rehabilitation monitoring application

1.2.5 Why are sensor networks different?

In this section, we presented four main application domains. Plenty of other applications exist such as pollution monitoring, home automation, traffic monitoring, etc. It is impossible to give a complete list of applications as it grows as potential users become aware of this technology. From the above overview, we can notice that there may be huge differences from one application to one another. Indeed, each application has its very own characteristics and they affect the network design. From this observation, we foresee that it is unlikely that there will be a "one size fits all" solution for all the possible applications. Besides, due to severe resource constraints and singularity of wireless sensor networks, traditional protocols of the TCP/IP stack may not be well suited. To each application corresponds a specific WSN and specific challenges thus specific protocols.

1.3 Application requirements and constraints

Although there is a wide variety of envisioned applications for WSN, most of them share some basic characteristics. In particular, they show common interaction or communication patterns. A common categorization consists in classifying WSN applications according to their data delivery model. We distinguish four different data delivery models: the periodic measurements model, the event-driven model, the query-driven model, and the hybrid model.

Periodic measurements In periodic measurement applications, nodes periodically collect and transmit readings to the sink, either directly or via multiple relay nodes (multiple hops). Sensor nodes are programmed to perform periodic measurements and reporting. The measurement period varies depending on the type of service that the application must fulfill (from one humidity reading per hour in agriculture to a sampling frequency of 50 Hz for an accelerometer), as well as the reporting period. The measurement and the reporting periods can be identical, for example when each sample is transmitted immediately. In order to reduce the number of transmissions, it is possible (if tolerated by the application in terms of delay) to set a longer reporting period, thus several samples are sent in one transmission.

Event-driven In event-driven applications, nodes monitor an area but don't transmit the collected data. Instead, alarms are triggered when events are detected (for example a motion sensor is activated, a temperature threshold is exceeded). The node can either report the detection or collaborate with neighbors to decide whether an event has occurred. Indeed, complicated types of event may involve several nodes. The detection of such events or the detection of several types of events within the same WSN require additional techniques such as event correlation and event classification. These tasks can be perform either by the sink or in the network by cluster heads or nodes. The source of the event can be mobile. WSN for tracking applications must not only be able to detect events but also estimate speed and direction as well. Typically, in this scenario, nodes have to cooperate before updates can be reported to the sink.

Query-driven In the query-driven (or user-initiated) model, the sensors only report their results in response to an explicit request from the user.

Hybrid Two or three approaches can coexist in the same network. The combination of the patterns form a hybrid model.

In addition to sharing communication patterns, wireless sensor networks may have other parameters in common. When designing a WSN, it is necessary to carefully study the characteristics and requirements of the target application, so that the network properly fulfills the application needs. We identified a set of technical parameters that must be considered in order to design an appropriate application for the intended monitoring task(s) and an efficient network. Below, we enumerate these parameters and we give a short overview of their impact on the network design. We grouped these parameters into two classes: application requirements and constraints.

Requirements The application requirements reflect the needs of the end user, how he expects the system to work. The application and the network protocols must be designed accordingly.

- Data precision: data precision is determined by the accuracy of the sensing hardware and also by the sampling frequency of the readings. If the sampling rate is two low, the information at the receiver end may not be detailed enough. Data precision also implies the correctness with which events are detected. Events must not be missed, and there should not be any false detections.

- Availability: in WSN, nodes can fail or die, when deployed in hazardous environment or when the battery is depleted. Some critical systems, for example, health care applications, require high availability. It is necessary to implement resilient applications and network protocols that ensure service continuity.

- Lifetime: wireless sensor network applications are intended for specific monitoring tasks. The application must operate as long as the user needs to observe the phenomena. According to the available energy resources, low power mechanisms may be implemented in order to achieve the envisaged application lifetime.

- Quality of Service (QoS) parameters: in sensing applications, the observer is interested in monitoring phenomena under some latency and reliability restrictions. QoS requirements usually come from demanding applications such as multimedia applications, but also from critical and delay-intolerant applications like boarder control and health care applications. In order to provide a satisfying user experience, the network must provide QoS support.

Constraints There are many constraints to consider when implementing a WSN application. They often are consequences of the nature of the phenomena to be observed, or of the environment in which the nodes are deployed.

- Size of the area to monitor: it gives the minimum number of nodes needed in the network. This factor also indicates if multihop communication is required.

- Coverage: the coverage of the monitored area can be either sparse or dense, full or partial. A dense coverage provides redundancy, adds data precision and can be used in order to improve the fault-tolerance. However, data redundancy may also consumes extra energy and bandwidth. The density of the network should be determined according to application requirements and the cost of sensor nodes.

- Type of phenomena to monitor: it has a strong influence on the node hardware. It determines what type of sensors are to be used (e.g., temperature sensors, motion sensors, cameras, etc.). It also have an influence on the size of the node, especially if the node is to be put on a human or an animal. According to the environment where to nodes will be deployed, additional features may be required, such as impermeability to water and dirt.

- Energy resources: in most of WSN application scenarios, it is not possible to connect nodes to an unlimited power source, for example, when nodes are deployed in hazardous or unreachable areas. Nodes will have to operate only with limited, sometimes irreplaceable, batteries. In order to meet the application lifetime requirements, the batteries must be correctly dimensioned. Power harvesting mechanisms may help to improve the lifetime of the batteries.

- Mobility: in some WSN architectures, some nodes could be mobile, for example, when attached to animals or deployed in a flowing river. Mobility must be taken into consideration so that it does not raise any communication problems.

- Deployment: the deployment of the nodes over the area to monitor can be either predicted or random. For the latter option, self-organization is required.

- Cost: the available budget is a concrete constraint which is nonetheless essential. Depending of the required type and number of sensors nodes, the cost of the application varies considerably. Trade-offs must be found between data precision, coverage and availability. Moreover, nodes programming and deployment involve an additional expense.

WSN are specifically designed for particular monitoring applications. They have to fulfill the application requirements while coping with numerous constraints. As many parameters are interdependent, the design of efficient WSN raise numerous challenges. As an example, it may be hard to provide high availability while using unreliable radio communications or to maintain the connectivity of the network in case of node failures. We further investigate WSN research challenges in Section 1.5. In the next section, we give go into the details of sensor network design, architecture, and protocols.

1.4 Sensor networks design, architecture, and protocols

1.4.1 WSN topologies

The size of WSN varies from a few nodes to several hundreds or even thousands. Accordingly, the topology of a wireless sensor network can vary from a simple star network to an advanced multi-hop wireless mesh network. We briefly discuss three basic types of wireless sensor network topologies.

Star network The star topology is the simplest WSN topology. It is used mostly when there are few nodes in the network. It requires that all nodes in the network are within the radio range of the base station. They directly transmit the gathered data to the sink without needing to communicate through other nodes. An example of a star network is shown in Fig. 1.6.

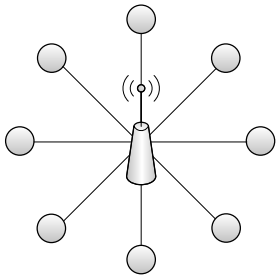


FIGURE 1.6: Star network

Tree network A tree network is a a hierarchical structure rooted at the sink. Branch nodes (nodes that have child nodes) are denoted as cluster heads. Usually, cluster heads are one level down from the root (they are directly connected to the sink). Leaf nodes (nodes that do not have child nodes) communicate with the base station through their cluster head. Tree networks are particularly useful when the area to monitor consists of several unconnected areas. An example of a tree network is shown in Fig. 1.7.

Mesh network Mesh networks are the ad-hoc topology of large WSN. When a node far from the sink (not within its radio range) has data to send, the data has to hop from node to node until it reaches its final destination. Each node collect and send data but also has to cooperate in order to relay data from other nodes. There is no hierarchy, the network must self-organize: construct the topology and implement routes to the sink. This type of networks is the most complex. It raises challenges such as how to find the best routes to the sink and how to balance the traffic not to empty the battery of a node. An example of a mesh network is shown in Fig. 1.8.

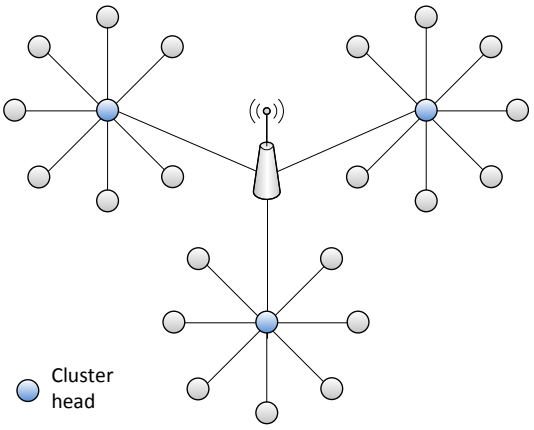


FIGURE 1.7: Tree network

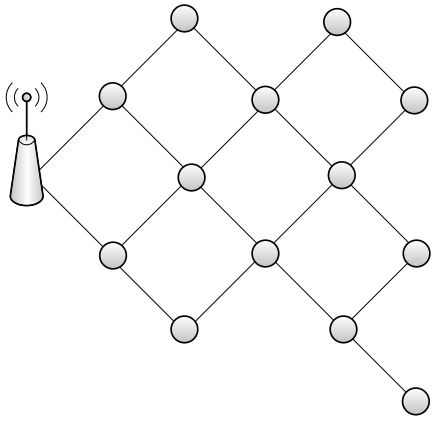


FIGURE 1.8: Mesh network

1.4.2 Communication protocols architecture

The sensor nodes communication protocol stack is organized similarly to the TCP/IP protocol stack, with the addition of three transverse management planes: the power management plane, the mobility management plane, and the task management plane. Fig. 1.9 illustrates the various layers of the wireless sensor networks protocol stack along with the management planes. The power management plane aims to minimize the overall energy consumption, the mobility management plan handles the network dynamics (movement, death and arrival of nodes), and finally the task management plane helps the sensor nodes to coordinate the sensing tasks. We give a short overview of the role of each layer following a bottom-up approach.

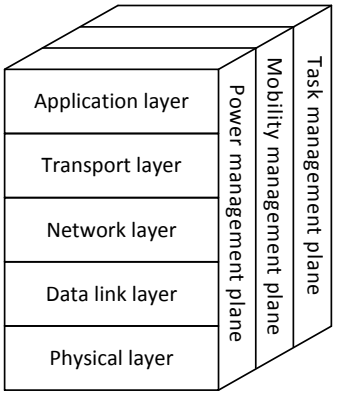


FIGURE 1.9: The sensor networks protocol stack [1]

Physical layer The physical layer performs the actual wireless communications (transmission and reception) between individual nodes. As WSN require long-range, low-power communications, radio frequency (RF) communications are preferred over other wireless media such as optical communications and ultrasound. In addition, RF communications provide acceptable error rates and does not need line of sight between sender and receiver. Communication requires a transmitter and a receiver. A transceiver is a device that combines these two tasks in a single entity. Transceivers design, modulation and coding schemes determine the maximum transmission distance, the available bandwidth, and greatly influence nodes lifetime, as energy consumption is due to a large extent to radio communications (we further investigate energy consumption in Chapter 2). Besides data transmission and reception, the physical layer is also responsible for activation and deactivation of the transceiver, channel frequency selection (through energy detection within the current channel and link quality indicator of received packets), and clear channel assessment (CCA). The CCA function allows the MAC layer to be aware of the medium state: busy or idle.

Legacy RF technologies such as Wi-Fi and Bluetooth technologies are not the most appropriate for developing a WSN. Bluetooth is limited to short-distance communications and Wi-Fi lacks a low-power mode. IEEE 802.15.4 is a protocol for low-rate wireless personal area networks (WPAN). This standard includes both physical (PHY) and MAC layer specifications.

The standard specifies four physical layers: three in the 868/915 MHz frequency bands and one in the 2.4 GHz frequency band. The latter is best suited as it is the only one to operate in a frequency band authorized worldwide (each country is responsible for its own spectrum management) and it offers the highest data rate. Indeed, a 2450 MHz direct sequence spread spectrum (DSSS) PHY employing offset quadrature phase-shift keying (O-QPSK) modulation allows a data rate up to 250 Kb/s.

Data link layer The data link layer is divided into two sublayers: the logical link control (LLC) sublayer and the media access control (MAC) sublayer. The LLC layer mostly provides error management mechanisms. The MAC layer handles all access to the physical radio channel. In radio communications, the transmission medium is shared among all nodes within radio range. Only one transmission may occur at a time. Two or more simultaneous transmissions would cause a collision: both transmissions get mixed up and not readable by the receiver. MAC protocols have to determine who is allowed to access the media while avoiding collisions. We identify three approaches in MAC protocols design: contention-based, contention-free, and hybrid.

Contention-based or random access protocols exploit randomness in order to minimize the collision probability. Protocols of this category are fully distributed, i.e., there is no preliminary coordination or organization, they tries to ensure that nodes do not transmit at the same time. This approach may lead to collisions. A typical contention-based MAC protocol is Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). When a station wants to transmit packets, it has to sense the medium in order to determine if there is an ongoing transmission or not. If the medium is not busy, the transmission may proceed. Otherwise, the station must defer its transmission. After a transmission attempt, either successful or unsuccessful, the station may wait for a random time, called backoff, before starting another transmission attempt. The random backoff procedure decreases medium contention conflicts when multiple stations are waiting for the medium to be available again.

Contention-free protocols divide the available resources between contenders such that each node can use its resources exclusively without the risk of collisions. Nodes must agree on a schedule which coordinates channel access among multiple nodes/users. The schedule is established using the exchange of signaling messages. Signaling mechanisms are also used to renegotiate the schedule, in case of a topology change for example. Two common contention-free protocols are Time Division Multiple Access (TDMA) and Frequency Division Multiple Access (FDMA). The TDMA protocol principle is to assign to each node a specific time period in which it has exclusive access to the medium. The TDMA scheme requires time synchronization. In FDMA, the available frequency band is divided into subchannels. Non interfering subchannels are assigned to nodes in the same neighborhood so they can carry out parallel transmissions on their channel. However, the receiver must be able to switch to the channel used by the transmitter. The FDMA approach requires a more complex transceiver.

Hybrid MAC protocols combine both approaches in order to enhance their strengths and offset their weaknesses. As an example, we can cite the slotted CSMA/CA algorithm. In this improved version of the CSMA/CA channel access mechanism, the time is divided into units denoted as superframes. Within the superframes, contention occurs just like in classical CSMA/CA, except that all nodes start the backoff procedure at the same time (at superframe boundaries) so the probability of collisions is reduced. This protocol needs a central coordinator which send synchronization messages called beacons at the beginning of each superframe. In addition, the superframe structure can have an active and an inactive portion. During the inactive period, nodes may sleep in order to save energy.

The IEEE 802.15.4 standard is an interesting example of wireless sensor networks MAC protocols, as it implements the three approaches. Different operation modes are available, depending on the network topology and requirements. The slotted CSMA/CA algorithm suits star and tree topologies, in which leaf nodes send the gathered data to central entities. The base station and the cluster heads can act as coordinators which define the superframe structure and send the beacon messages. This mode combines time division and contention, therefore its operation is hybrid. For low-latency applications, there is an alternative version of this protocol where a part of the active period of the superframe is reserved for low-latency traffic. This part is called CFP, as contention free period, and consists of time slots reserved for priority nodes. Actually, during the CFP, the protocol behaves like TDMA. Finally, in peer-to-peer or ad-hoc topologies, if the network does not require synchronization or support for low-latency devices, the 802.15.4 MAC uses classical CSMA/CA.

Network layer In a multihop wireless sensor network, some source nodes are located at more than one hop from their destination and cannot reach it directly. The packets must be send through intermediate nodes, which have to relay the received packets to the final destination (generally the base station). Intermediate nodes must know where to forward an incoming packet not destined for itself; the purpose of the network layer is to organize end-to-end packet delivery. It issues service requests to the data link layer, which is responsible for hop-to-hop delivery. The network layer performs network routing, i.e., it assigns a network address to each node, constructs paths, and maintains the routing table. The routing table references where to forward packets according to their destination. Usually, in WSN, the destination is the base station. The routing protocol computes a cost for each possible path, so when several next hops are available, the route having the lowest cost is preferred. Several approaches to evaluate the best path have been proposed, depending on different strategies, such as shortest path, highest energy path, highest bandwidth path, lowest delay path, etc. Fig. 1.10 gives a simple example of routing where the source node S wants to send packets to the sink. Table 1.11 shows an excerpt of relay node A's routing table, using the hop count as cost metric. It contains two routes to the sink: through node B or node C. The selected route is via node B (represented by the solid lines) since it has a lower cost.

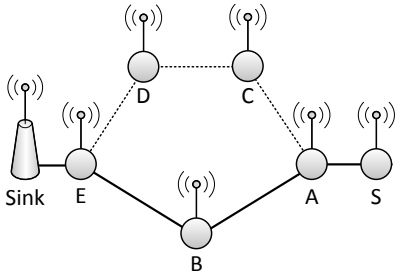


FIGURE 1.10: Simple routing example

Destination	Next hop	Cost
B	B	1
C	C	1
Sink	B	3
Sink	C	4
...	...	...

FIGURE 1.11: Node A's routing table

The construction and the maintenance of routing tables require the exchange of signaling messages. As transmission consumes a lot of energy, it is necessary to optimize the number of routing messages. For instance, reactive protocols only construct routes when a packet is to be sent to an unknown destination, so the number of signaling messages used is minimized. However, it may take a long time to find a path to the destination, thus degrading packet latency. For real-time applications, it is necessary to develop routing protocols that offer a good tradeoff between latency and energy consumption. Proactive protocols build the routes in advance and try to keep accurate information in the routing tables. Other routing protocols combine both approaches an are denoted as hybrid. Routing protocols can be classified according to these

three categories, depending on the method used for constructing paths: reactive, proactive, and hybrid. Another common classification method of routing protocols depends on the underlying network structure and divides them into flat, hierarchical, and geographical. In flat routing protocols, all nodes play the same role. They are used in large networks where nodes collaborate in a global sensing task. The hierarchical routing concept allows to improve the network lifetime. This category of routing protocols suits the tree topology, where the cluster heads are special nodes with higher energy that can be used to handle higher loads and perform data aggregation in order to reduce the number of redundant data packets. In geographical or location-based protocols, the nodes exploit their location information rather than message exchange to construct paths. This technique assumes that nodes are able to determine their location and to compute the distance to neighboring nodes.

Transport layer The role of the transport layer is to maintain the flow of data (sequence of packets) if the sensor network application requires it by implementing end-to-end data recovery and congestion control mechanisms. It may help the sender to adjust the packet rate in order to avoid congestion and buffer overflow. In addition, transport protocols may detect packet loss at the receiver side and trigger retransmission. Reliability is the major concern of transport protocols, and it is a major requirement in many WSN applications. However, in WSN, there are no strict end-to-end data flows, but nodes rather collaborate to detect physical phenomena thus the data concerning a specific event detection may come from several sources. Therefore, using transport layer protocols may not be relevant in WSN. Nevertheless, reliability techniques for WSN exist, although they are not placed exactly at the transport layer but are rather cross-layer and are not referred to as transport protocols. In WSN, it is more appropriate to define the reliability as how reliably the WSN application detects the monitored phenomena, instead of the percentage of packets delivered (although it may be related). In many applications, the reliability simply relies on redundancy, typically by ensuring that several sensors would be activated when the monitored events occur. Nevertheless, when it is not possible to deploy several sensors to detect an event such as the fall of a patient at the hospital, loss recovery mechanisms are required.

Application layer The application layer is the top level of the communication architecture and interfaces directly with the application. Although many application fields for sensor networks have been defined and many applications proposed, the contributions in this area are marginal compared to the profusion of research work focusing on physical, MAC and network layer

issues. As many applications share some characteristics, application protocols providing high-level interfaces would facilitate applications implementation. Application protocols are needed in WSN, just as in traditional networks: protocols for addressing, time synchronization, location services, interfaces to query the data which the user is interested in, etc.

1.5 Sensor networks challenges

In Section 1.3, we identified characteristics and requirements of wireless sensor network applications, then we gave a short overview of the communication architecture and protocols of WSN in Section 1.4. Given the severe resource constraints of sensor nodes and the rigid layered architecture inherited from the design of the Internet, it is clear that fulfilling the application requirements will be a complex task. In the following, we highlight the main technical problems and challenges when designing efficient wireless sensor networks.

1.5.1 General design considerations

Energy and network lifetime As sensor nodes are powered with batteries in most applications, energy is a major concern. An adequate WSN must perform its intended function during the required duration, otherwise it is vain. To ensure an appropriate network lifetime, all protocols of the communication architecture have to be energy aware. Since the transmission of messages represent an important part of energy expenditures, it is necessary to design algorithms/protocols that induces few signaling messages. Also, if admissible by the application, nodes may go into a low-power (sleep) mode during periods of inactivity or according to a predefined schedule. The topology plays an important role concerning the network lifetime. Some nodes may be involved in many communication paths (typically nodes close to the base station), so their energy decreases faster than that of nodes which do not act as relay nodes. As a consequence, the network may be down although almost all nodes are working properly, just because a few strategic nodes fail leading to the network being disconnected. In addition to reducing energy expenditures, sensor nodes may be designed to harvest energy in order to prolong the battery life. This can be achieved by employing solar cells, using vibration-based technologies or thermoelectric devices. It is impossible to provide an exhaustive list of techniques to minimize energy consumption. The previous examples only point out some levers. A survey on energy conservation in WSN is provided in [3].

Scalability Since many WSN applications require a large node count, communication protocols should be able to scale. Scaling also refers to the ability of the system to adapt to an increased traffic load, handling and processing large amounts of data. Less scalable protocols may lead to severe performance degradation (for example increased latency and packet loss). Distributed algorithms may be preferred over centralized solutions. An illustration of a challenging task in big networks is time synchronization.

1.5.2 Communication architecture challenges

Quality of service New applications for WSN with high quality of service requirements have recently emerged. Quality of service refers to the level of performance of the application and translates into network parameters such as delay and reliability. Applications with high QoS requirements have strict delay, reliability or bandwidth requirements, e.g., multimedia and health care applications. In order to provide QoS support, protocols have to implement specific, often energy consuming mechanisms. As energy efficiency is a key element in wireless sensor networks design, QoS communication protocols should balance energy efficiency and optimization of the QoS parameters. QoS provisioning in WSN is studied in depth in the next chapter.

Mobility support In wireless sensor networks, nodes can be static or mobile, depending on the application requirements. Mobility may be seen as an issue and also as an opportunity. One one hand, dealing with mobility can pose some formidable challenges in protocol design, but on the other hand, mobility can enhance the operation of WSN by extending its capacities. For example, mobile entities can deploy sensors in order to re-establish lost connectivity in the network, perform energy harvesting, or enhance the localization capabilities of sensors [4]. Also, in some scenarios, it may be more energy efficient to have a mobile sink collecting data rather than forwarding the data hop by hop. MAC and routing protocols design must handle topology changes such that mobile nodes are able to communicate just as fix nodes. Nodes arrival and death are also a form of mobility: new nodes must be able to join the network and it is necessary to overcome the loss of nodes. Efficient algorithms must be designed to guide robots to track objects or reconnect the network and optimize the route of a mobile sink which needs to visit all fix nodes.

Internetworking One of the most attractive features of WSN is to manage the network from the Internet. However, it triggers the need for a gateway in order to link IP with the network protocol used inside the WSN. This also applies for mobile and satellite communications. The design of such gateways is an open research issue.

Security Since WSN are envisioned for critical applications such as heath care monitoring, fault-tolerance mechanisms are needed. Several types of failure may happen, as well as attacks. Attackers may try to bring the network down or steal critical or personal data. Fault-tolerance, security, and privacy are challenging issues considering the limited capabilities of sensor nodes and the distributed nature of WSN. For instance, in order to achieve privacy, confidential data may be encoded. Data encryption and decryption takes time and consumes energy.

1.5.3 Additional issues

Localization The purpose of localization techniques in WSN is to perform accurate localization of nodes. Indeed, several WSN applications require the correlation of the sensor readings with physical locations, for example to build maps of the recorded events. Some applications that rely on localization would not be permitted without localization techniques, such as target tracking applications. In addition, localization may benefit to various networks services like location-aware routing, data aggregation, etc. The location information may be provided to the nodes in several ways: it may be set during the node configuration, acquired thanks to sensors, or calculated by collaborating with other nodes. Since in large networks, it may not be possible to manually set the location of all nodes, and having GPS hardware on every node can be very costly (both in terms of money, energy consumption, size, etc.), it is necessary to design efficient and energy-aware localization algorithms. We can classify the localization algorithms into the two following categories: centralized and distributed. Centralized algorithm are efficient but not scalable. Distributed algorithms require anchor nodes which location is preloaded or obtained with a GPS, then standard nodes computed their relative position using methods such as trilateration. The distance with anchor nodes can be estimated with time or signal information, for example with the Received Signal Strength Indicator (RSSI). Such methods are quite cheap but not very precise. Designing accurate, low-cost, scalable, and energy efficient solutions is a challenging task.

Heterogeneity support While in some WSN applications all nodes are identical, other may require nodes with different sensing capabilities or energy resources, or even involve other types of hardware such as RFID tags. Supporting heterogeneity implies that the network should operate efficiently despite the device diversity. In order to achieve this, heterogeneity has to be turned into an advantage: routing protocols may exploit nodes with higher energy resources in order to prolong the network lifetime, nodes with higher memory and computation resources may act as cluster heads or aggregation nodes, etc. Heterogeneity also refers to traffic heterogeneity. Different types of data having their own requirements may coexist in the network. In order to support these different types of traffic and fulfill their requirements, the network must be able to differentiate them and behave adequately regarding their specific requirements.

1.6 Conclusion and prospects

In this chapter, we covered the basics of wireless sensor networks. We showed the number and variety of envisioned WSN applications, then we discussed the specific characteristics of WSN and why traditional protocols cannot be used. During this introductory study, after having presented the communication architecture, we listed design open issues and research challenges. Since the beginning of research efforts in the field of WSN, research efforts concentrate on providing energy-aware communication protocols that maximize the network lifetime. Recently, new applications have been proposed with high requirements, and protocols designed for low-rate, single-purpose applications may not provide an appropriate level of service for this new category of applications. New mechanisms have to be found in order to allow the requirements of these applications to be fulfilled. Due to the severe resource constraints of WSN, QoS support for WSN applications is a complex and challenging issue. In this thesis, we focus on quality of service provisioning for multimedia and high demanding WSN application, especially at the MAC layer. Our research interests also include heterogeneity support. In the remainder of this manuscript, we give an overview of QoS and heterogeneity support for WSN applications with high requirements, and we expose our contributions along with extensive performance evaluations of the proposed solutions.

Chapter 2

Quality of Service and Wireless Sensor Networks

This chapter describes problems and existing solutions of quality of service provisioning in wireless sensor networks. First, we introduce our definition of Quality of Service (QoS), then we discuss QoS provisioning approaches in traditional data networks and QoS support in wireless sensor networks. Finally, we assess the state of the art on QoS support in WSN at each layer of the protocol stack and we highlight the remaining research challenges open issues.

2.1 Overview of quality of service

2.1.1 Definition of QoS

Depending of the point of view, the term QoS may have different meanings. The locution QoS may refer to the application requirements and to the degree to which the system performs its intended functions, but also to the mechanisms implemented to provide this performance. There are two perspectives, user-oriented and network-oriented. These two perspectives are interdependent as shown in Fig. 2.1: applications and users have QoS requirements, and the network must provide QoS support. They are two sides of a single concept. Since our work focuses on providing efficient QoS support for wireless sensor networks, we naturally consider the concept of QoS from the network-oriented perspective. In the following subsections, we are going to explain in detail the concept of quality of service as seen from the network side by analyzing common network issues and how they affect the network performance. Afterwards, we introduce some QoS control mechanisms which are designed to overcome these issues.

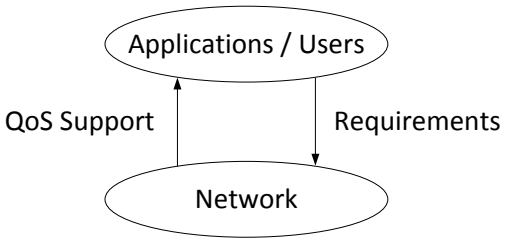


FIGURE 2.1: QoS interdependence [2]

2.1.2 Factors affecting the quality of service

Some applications like multimedia and critical applications have high requirements such as high availability of service, stability of service, and low delays. Basically, these user-oriented requirements translates into packet latency, throughput, and reliability concerns. If the network is not able to provide an appropriate support to these demanding applications (adequate latency, sufficient throughput and high reliability), the user experience can be degraded, or the application can even become ineffective. Data networks carry data packets from a sender to the intended receiver. As packets travel from source to destination, several problems may happen, affecting packet delivery and decreasing the level of QoS.

Errors Packets can be corrupted due to noise and interference, especially in wireless communications. The receiver has to detect if the packet is corrupted and try to recover the initial packets (if correction mechanisms exist). If the recovery is unsuccessful, the packet is dropped.

Low throughput Due to the concurrent flows of other users sharing the same network resources, the maximum throughput that can be provided to a given stream may be too low, particularly for high bandwidth consuming applications, such as audio or video streaming. Moreover, when an intermediate node receives more data than it can transmit, it results in network congestion. The buffer fills up, and once it is full, all packets received thereafter are discarded.

These problems may have heavy consequences at various levels.

Packet loss Dropped packets affect the application quality and reliability. In applications like multimedia streaming, the loss of little data can be tolerated. However, many applications are not loss tolerant (e-mail for example) and mechanisms must provide reliable end-to-end delivery for these applications.

Delays Latency is the time elapsed between when a source initiates the sending of a packet and when the packet is received. It is the combination of propagation, queuing, and processing delays. High network load may cause increased queuing delays. As for packet loss, applications may be more or less delay tolerant, but the important thing is to deliver the data before it gets outdated. Packets that miss their deadline are no longer meaningful and may be dropped.

Jitter Jitter refers to the packet delay variation, i.e., the difference in source to destination delay between packets in a given flow. Jitter is an important issue in real-time applications. The effects of delay variation can be compensated using a buffer at the receiver end, at the cost of an additional delay.

Out-of-order delivery Sometimes the packet delay variation is such that the packets arrive in a different order than they were sent. Reordering mechanisms are needed, causing additional queuing and processing delays. Applications as video and voice over IP (VoIP) are dramatically affected by both latency, jitter, and lack of sequence.

2.1.3 QoS control mechanisms

In the previous section, we identified common network issues and how they affect the performance of the network. QoS mechanisms are needed in order to overcome these issues. Several techniques were proposed. In what follows, we study various QoS control mechanisms which aim to improve latency and reliability, and also to optimize network resource utilization, such as bandwidth. As these techniques relate to various layers of the communication protocol stack, the reader may refer to the state of the art provided in Section 2.4 in order to get an overview of how these mechanisms are implemented in QoS protocols at each layer of the protocol stack.

Scheduling Scheduling is used for transmitting multiple flows simultaneously while distributing resources. The main purposes of scheduling algorithms are to maximize throughput and reduce delays while avoiding resource starvation and ensuring fairness among the data flows or achieve QoS goals. Several scheduling algorithms have been proposed in the literature. The simplest algorithm is known as first come, first served (FCFS). First in first out (FIFO) queues are simply processed in order, no prioritization occurs. In strict priority (SP) scheduling, a fixed priority is assign to each data flow, and packets in the queue are processed in order of their

priority. However, starvation of lower priority traffic may happen in presence of large amounts of high priority traffic. With the weighted fair queuing (WFQ) technique, different scheduling priorities are assigned to data flows. Each data flow is assigned to a separate FIFO queue, and each queue is assign a weight. Since queues are served in proportion to their weight, WFQ allows to finely allocate the available bandwidth according the requirements of each flow. Fairness between all flows can be achieved by setting an equal weight to each queue. Earliest deadline first (EDF) strategy arranges packets with the closest deadline to be next in the queue. This requires advanced knowledge or estimations about the time required for a packet to arrive to its destination as well as sorting algorithms. Such a mechanisms provides low latency for real-time traffic.

Rate limiting Rate limiting consists in controlling the rate of traffic sent by a source or transiting by intermediate nodes. Traffic shaping and flow control are two common rate limiting techniques. Traffic shaping aims to improve latency and/or throughput of high priority data flows by delaying other kinds. The leaky bucket and the token bucket are two popular traffic shaping algorithms. The leaky bucket algorithm is based on a analogy of a bucket with a hole in the bottom through which water leaks away at a constant rate. If the incoming water rate is larger that the leaking rate, the water will exceed the capacity of the bucket which will overflow. The token bucket algorithm is based on a similar analogy using tokens. Flow control is another form of rate limiting which prevents the sender from overwhelming the receiver and from overloading the network. The sender is informed, either by acknowledgments sent back by the receiver or after timeouts, of how it has to adjust its sending rate. The most common flow control algorithm is TCP rate control. This end-to-end flow control protocol avoids ending-up in the situation where the sender sends data too fast by using a sliding window mechanism. The window size is continuously adjusted to maximize the throughput.

Congestion avoidance Traffic shaping and flow control are part of preventive methods to avoid network congestion, nevertheless the implementation of these mechanisms may not always be sufficient to elude network congestion. In particular, flow control algorithm often adjust the sending rate in response to packet loss, which means that congestion is already effective. The goal of congestion avoidance techniques is to prevent network congestion by monitoring congestion symptoms such as packet loss, delays, buffers filling ratio, and by triggering control

mechanisms. Random early detection or random early drop (RED) is an active queue management algorithm which, instead of using the conventional tail drop algorithm, drops incoming packets based on statistical probabilities. The probability to drop a packet increases as the buffer fills up. This mechanism allows to trigger flow control mechanisms before the network is fully congested. A similar technique is called explicit congestion notification (ECN). ECN-aware routers may signal congestion to traffic sources instead of dropping packets. Flow control and congestion avoidance mechanisms are typical mechanisms of the transport layer.

Packet loss remediation Reliability is a critical requirement, however it is not easy to guarantee the delivery of packets. The most commonly used method which aims to overcome packet loss is to acknowledge receipt of packets, so when a sender does not receive an acknowledgment back after sending a packet, it is informed that the packet has not been received and may resend the lost packet. However, this method cannot guarantee the delivery of a packet. If the transmission keeps failing, the sender will stop trying to send the packet after having exceeded the maximum number of allowed retransmissions and the packet will be dropped. Acknowledgments are mainly implemented at the transport layer to acknowledge end-to-end packet delivery and also at the MAC layer in wireless networks, as the medium is unreliable. Another technique which aims to improve the reliability is to maximize the probability that packets are successfully delivered by introducing or increasing data redundancy in the network. A source may send multiple copies of the same packet, preferably over different routes, to ensure that the data reaches its destination. Nevertheless, this technique requires that a mechanism that handles duplicate packets be implemented at the receiver. Such a technique may also improve latency, since the probability that the packet will be delivered at the first attempt is higher, but it consumes extra bandwidth.

QoS control mechanisms attempt to overcome some network issues in order to improve the network performance. While analyzing common network issues and their consequences, we can see that most of the time, issues arise as the network load increases. Many mechanisms presented in this section aim to optimize network resource utilization and to prevent network overload. Resource management is the key issue of QoS support. If the network capacity is sufficient, it eliminates the need for QoS mechanisms. On the contrary, QoS mechanisms are required to ensure the proper operation of applications with high QoS requirements in case of peak traffic loads or under heavy resource constrained conditions.

2.2 QoS provisioning approaches in traditional data networks

In this section, we briefly introduce QoS provisioning approaches in traditional wired and wireless IP networks.

2.2.1 The transport layer: a first step towards QoS

Initially, IP, which stands for Internet Protocol, was not designed to be reliable and is a best effort delivery protocol. The transport layer, situated between the application layer and the network layer, provides end-to-end communication services for applications. According to applications requirements, transport layer implementations will determine whether or not to provide reliability. Two implementations are provided in traditional networks: Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). UDP is a simple transmission model which does not provide reliability because in time sensitive applications, dropping packets is preferable to waiting for delayed packets. Nevertheless data integrity is provided using checksums. TCP provides both data integrity and reliable data delivery. A connection is established between the sender and the receiver. The packets are identified by sequence numbers. These numbers are used at the receiver to order the received packets and to communicate to the sender in ACK messages which packets have been received. TCP also implements flow control and congestion control mechanisms.

2.2.2 Integrated approaches

The transport layer is a first step towards QoS but is it not enough to satisfy the requirements of critical applications over less demanding applications. The network must be able to differentiate between the different data flows in the network and to provide an appropriate level of QoS to each flow. Integrated approaches are needed. Two QoS provisioning approaches exist: the hard QoS provisioning approach and the soft approach. The hard QoS provisioning approach aims to guarantee a minimum level of QoS using resource reservation. The soft QoS provisioning approach defines priority levels for different traffic types without however providing a strict guarantee as to a minimum level of performance. In the following, we discuss two common architectures which illustrate each approach.

IntServ IntServ [5] is a well-know example of hard QoS architecture. The idea is that for every flow that requires some kind of guarantees, the application has to make an individual reservation. Two underlying protocols are used to describe what the reservation is for (Flow Specs) and to signal the reservation across the network (RSVP). IntServ is an efficient mechanism as it provides efficient QoS guarantees through end-to-end bandwidth reservation. However, in order for IntServ to operate, all nodes along the traffic path must support it. In addition, IntServ is not scalable. In large networks, reservation requests add up and routers may to be able to admit all bandwidth reservations, and it is difficult to keep track of all reservations.

DiffServ Differentiated Services or DiffServ [6] is a soft QoS architecture introduced to meet the demands for QoS of the Internet. The DiffServ approach operates on the principle of traffic classification, ensuring preferential treatment for higher-priority traffic classes. Packets are classified and marked (or tagged) at the edge of the network, then specific forwarding treatments, formally called Per-Hop Behavior (PHB), are applied on each network element, providing the packet the appropriate delay-bound, jitter-bound, bandwidth, etc. In DiffServ, there is no need for signaling and keeping track of reservations. The combination of packet marking and well-defined PHBs results in a scalable QoS solution.

2.2.3 Wireless networks

QoS challenges in wireless networks arise from lower bandwidth, unreliable medium and mobility support. Considering these specific issues, QoS support is particularly desirable, especially for demanding application such as voice over wireless local area networks (VoWLAN). Since wireless local area networks (WLANs) are an extension of wired networks for mobile users (e.g., laptops and PDAs), it is natural to integrate the QoS architecture deployed in wired networks with wireless MAC protocols. The IEEE 802.11e standard [7] is an extension of the Wi-Fi protocol suite and specifies QoS enhancements at the MAC layer that are consistent with DiffServ. It provides differentiated access control to handle the various QoS requirements of simultaneous applications or users. The standard defines four priority levels called access categories (ACs): voice, video, best effort and background, that can be mapped to DiffServ classed of service and reciprocally. Higher chance of being sent is given to high priority traffic over low priority traffic using various mechanisms. For example, voice and video ACs contention window is smaller than that of best effort and background traffic, so they have higher chance of

accessing the channel. In addition, they may benefit from contention-free access to the channel for a period called a Transmit Opportunity (TXOP). During this interval, a station can send as many packets as possible.

Since advances in technology (e.g., optical fiber and MIMO antennas) continuously make the network capacity increase (in terms of bandwidth, memory, processing capabilities, etc.), over-provisioning of network resources remains the most simple and the most widespread way to provide QoS support. However, at the same time, applications are also more and more demanding (as online video games, HD multimedia streaming). Supporting QoS is still an important issue and it has to be further developed for future applications, in traditional data networks as well as in emergent networks such as wireless sensor networks.

2.3 QoS provisioning approaches in WSN

Sensor networks are designed for different monitoring tasks but they all aim to detect events both reliably and timely, i.e., within the required time bounds. However, applications have various requirements.Fig. 2.2 shows a classification of traffic types according to their reliability and delay requirements/characteristics. This model can be also completed with a third dimension representing the amount of traffic, showing bandwidth requirements. We identified four types of traffic: delay-tolerant and loss-tolerant, delay-tolerant and critical, real-time and loss-tolerant, real-time and critical. Traffic classes with high latency or/and reliability requirements necessitate an appropriate support in order to satisfy these requirements and allow the application to function properly.

Wireless sensor networks are different from traditional wireless networks: they have specific challenges due to the limited resources. The available bandwidth in WSN is low compared to other wireless networks such as Wi-Fi and sensor nodes have low power, memory, and computation capabilities. In wireless sensor networks that simply rely on scalar data such as temperature or humidity readings and specialize in single-purpose applications, the traffic is mostly delay and loss tolerant. The network resources may be adequate to support the low requirements of these applications and they may not necessitate the implementation of specific mechanisms to ensure that the applications requirements are fulfilled. Nevertheless, it may not be sufficient to satisfy the requirements of more demanding and/or heterogeneous applications (applications with high requirements or having simultaneous different types of traffic). Unlike in traditional

data networks, over-provisioning may not applicable in WSN projects where the cost is a major constraint, as well as the size of the sensor nodes. Therefore, QoS support is essential for demanding and/or heterogeneous applications to operate as expected.

In Section 1.2, we listed a few application examples. From this study, we identified the most demanding applications and representative scenarios to demonstrate the need for QoS support. In what follows, we provide a brief overview of some of these applications which, in our opinion, are the most representative to demonstrate the potential of WSN and the difficulty of designing a robust system.

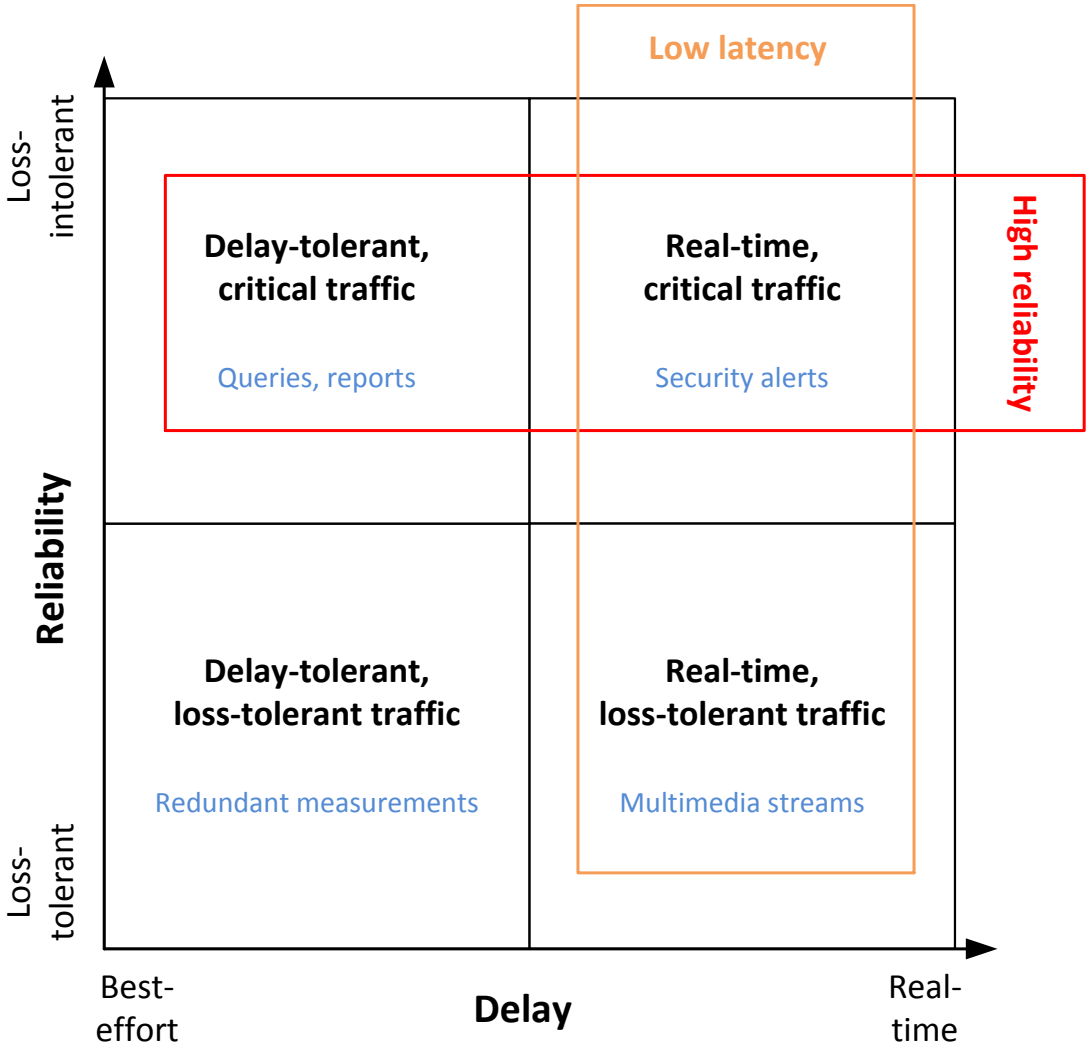


FIGURE 2.2: Classification of traffic types according to their characteristics

Real-time surveillance The aim of real-time surveillance applications is to secure a given area and to send alerts when the system detects potential threats. Typical real-time surveillance applications are border control, wildfire detection, battlefield surveillance, etc. These WSN are mostly event-driven, i.e., nodes react to the detection of unusual conditions and cooperate to decide whether an event has occurred and to report to the sink within a short time. For instance, in the wildfire scenario, nodes detecting an increase of the measured temperature must correlate their data with those of neighboring nodes in order to decide if there is a potential fire outbreak. They have to quickly send a report to the sink so that firefighters can intervene before the fire spreads. In such applications, the amount of traffic towards the sink is low but high reliability and short response times are required, as security is involved.

Tracking In tracking applications, the WSN system targets moving objects located in a given area, locates them, stores historical data, and obtains statistics. The system monitors the behavior of the targets and can detect abnormal situations. There are many scenarios in which tracking can be useful: biodiversity or patient monitoring, disaster response, and other location-based services. These applications need real-time information to allow the supervisors (scientists, nurses, etc.) to interact with the targets. The underlying system must provide low end-to-end delays, a certain level of reliability, and fault tolerance. In addition, since these applications collect and send relatively large amounts of data, WSN for tracking applications must ensure a fair sharing of the bandwidth. They also have to differentiate between best effort traffic and critical alarms and behave according to the characteristics of each traffic class in order to meet the delay and reliability requirements. Another challenge is to provide a reliable localization system. Finally, context-awareness and artificial intelligence mechanisms are needed to distinguish between the normal behavior of the targets and abnormal situations.

Multimedia applications Multimedia sensors can provide information-rich content enabling complex tasks such as identification and tracking. Wireless multimedia sensor networks (WMSN) will enhance existing sensor network applications and enable new application such as car traffic monitoring, control systems, etc. Indeed, visual information is undoubtedly the most desirable form of rich content. However, processing multimedia content in resource-constrained environments such as WSN is challenging. In multimedia applications, the data rate is high and the energy spent for communication raises accordingly, thus multimedia sensor nodes may require

the implementation of compression techniques in order to reduce energy consumption and improve the performance. In case of a redundant sensor deployment, information aggregation may help to reduce the amount of data to deliver to the sink. Nevertheless, complex in-network processing may expend as much power as the transmission of raw data. We distinguish two types of multimedia applications according to the delivery modes of multimedia content: snapshot and streaming. Snapshot content result from the detection of an event and contain observations obtained in a short period of time. Streaming denotes multimedia content generated over a longer time periods and having more strict delay requirements. Bandwidth and delay are the main concerns of multimedia content delivery. Reliability is considered as a secondary constraint, as multimedia traffic is relatively loss-tolerant.

Health care applications The medical applications of wireless sensor networks aim to improve the existing health care and monitoring services. They enable unobtrusive remote pervasive monitoring thus allowing patients to live a more independent and easy life [8]. Providing real-time and reliable data gathering and action taking are among the main benefits and challenges of WSN heath care applications. As life of human beings is involved, quality of service support in such systems is vital. Also, they must be able to identify the context. Emergency situations must be detected early and reliably. Data interpretation is a core issue of such smart applications. Various types of health care applications are envisioned, from simple BANs to large scenarios where one WSN perform the monitoring of patients and medical staff in a hospital. In health care applications, a large variety of data is collected such as localization data and vital signs. Alarms can also be triggered when emergency situations are detected. Since these different traffic types have their own QoS requirements, WSN for healthcare have to differentiate between the different classes of traffic according to their priority/criticality.

From this overview, we notice that several applications require QoS support to operate. WSN have to provide an efficient use of the limited resources to meet the QoS requirements of these applications. Their QoS requirements have to be identified and analyzed, then appropriate QoS mechanisms and QoS aware protocols must be designed and implemented to fulfill these requirements. Each type of data has its own reliability, delay, and bandwidth requirements. Wireless sensor networks combining several sensing modalities must handle different types of traffic with various characteristics, differentiate between them, and provide an appropriate level of QoS for each traffic class. Finally, although energy consumption is a major concern in WSN, especially in demanding applications, QoS and energy efficiency may result in competing interests in the

network design, therefore trade-offs must be found between QoS provisioning and energy savings. In conclusion, QoS support in WSN require the design of dynamic QoS mechanisms which adapt to the network conditions in order to optimize energy consumption.

2.4 Research efforts on QoS communication protocols for WSN

Until recently, supporting quality of service in WSN was still a largely unexplored research field. While a lot of research had been carried out on energy efficiency, only few studies had been done on service differentiation, real-time traffic or multimedia streams support, reliability, etc. This gap tends to be filled, however there is still a long way to go for the QoS requirements of complex networks like heterogeneous WSN to be fully supported.

In this section, we give an overview of research efforts on QoS provisioning in WSN. Several solutions have been proposed in order to fit the requirements of the different traffic types introduced in our classification shown in Fig. 2.2. We discuss contributions and shortcomings of existing work per layer and we point out remaining open issues. The role of each layer was briefly introduced in Section 1.4.2. The reader may refer to this section as a reminder.

Since our work in this thesis focuses on the MAC layer, an overview of specific MAC layer issues will be presented separately. We provide a more comprehensive state of the art on QoS provisioning for this layer in the next section.

2.4.1 Transport layer

Transport protocols are used to limit congestion, reduce packet loss and guarantee reliability while ensuring fairness in bandwidth allocation. Reliability support is an important concern since high reliability is highly desirable in many critical wireless sensor network applications. However, the traditional transport protocols TCP cannot be directly implemented for WSN due to several distinctive features of WSN listed below.

- *Many-to-one communication paradigm:* In wireless sensor networks, most of the traffic flows from several sensor nodes to one sink. The communication paradigm is not many-to-many like in the Internet but rather many-to-one. Due to this communication pattern, the amount of traffic grows as it gets closer to the sink. WSN are thus more prone to

congestion and would benefit from transport protocols. Also, as WSN are implemented in order to fulfill a specific mission, reliability is essential. Unlike in traditional networks, the transported data are events, not traffic flows from one source to a destination. Transport protocols for WSN should provide event reliability rather than packet reliability.

- *Resource constraints:* WSN have limited resources including small memory and computational capability, low bandwidth, and limited energy. TCP guarantees successful end-to-end transmission of packets at the cost of a high overhead (connection establishment, ACK, etc.) which is not necessary for event-driven applications.
- *Unreliable wireless medium:* TCP flow and congestion control mechanism relies on packet loss and it assumes that packet loss only originates from congestion whereas in wireless networks, the medium is unreliable and packet loss may be due to collisions or interference. Since TCP is not able to differentiate between the lost packets due to congestion or interference and triggers rate reduction whenever packet loss is detected, the implementation of TCP in wireless sensor networks may result in low channel utilization.

In addition to reliability and congestion control, an efficient transport protocol should consider fairness and QoS (latency and throughput) while limiting energy consumption. TCP is not energy-aware and does not support several traffic classes. Also, it may disadvantage nodes that are far away from the sink and result into unfair data delivery.

Loss recovery mechanisms and congestion control are independent features that can be provided separately or together. From the several transport protocols that have been designed for WSN, some of them have addressed either congestion or reliability only, while others consider both of them. Wang et al. [9] provided an overview of a dozen of transport protocols for WSN. In what follows, we first present some generic approaches for providing congestion control and loss recovery, then we discuss selected contributions.

Congestion control Congestion control protocols ensure congestion detection, congestion notification, and rate-adjustment. Some of them may detect congestion by monitoring the queue length while others look at packet service time. Then, transport protocols need to propagate congestion information to the sources that contribute to congestion. They either send explicit congestion notification messages (specific messages) or they use implicit congestion notification that consists of piggybacking congestion information in data packets. Finally, they implement end-to-end or hop-by-hop rate adjustment in order to mitigate the congestion.

Loss recovery Many WSN applications require the reliable delivery of events or packets. Reliable transport protocols must be able to detect when packets are lost or when the reliability drops, i.e., when not enough data reaches the sink, and to restore the lost data/events. Since in WSN, the transported data are mostly events and may originate from several sources, hop-by-hop loss detection and notification should be preferred over end-to-end approaches. Hop-by-hop approaches suits the traffic characteristics of event-driven WSN and are more energy efficient. There are basically three ways to notify the sender when packet loss is experienced: positive ACK, negative ACK, and piggybacked ACK. Loss notifications may also inform the sender of the reason for packet loss, as different countermeasures should be employed depending if the packet loss is caused by buffer overflow or interference. The reliability may be improved using retransmission or by increasing the source rate. For the latter mechanisms, it is required to distinguish between packet loss due to congestion or due to channel errors, because in case of congestion, increasing the source rate will not only be inefficient but may worsen the problem.

ESRT [10] is an event-to-sink reliable transport protocol. It aims to provide event reliability for data-centric WSN where end-to-end data reliability mechanisms are not appropriate. The algorithm mainly runs on sink. The base station computes the event reliability as the fraction of packets successfully received and informs the sources how to adjust the data rate in order to meet the reliability requirements of each event while not wasting unnecessary energy. Intermediate nodes monitor the congestion level: when a buffer threshold is exceeded, congested nodes set a congestion notification bit in the packet header so that the sink is informed. ESRT does not implement any retransmission mechanism: high reliability is provided only by adjusting the data rate of sources. As a consequence, it may not be suitable for WSN monitoring transitory phenomena. STCP [11] uses similar congestion control techniques but provides different reliability mechanisms according to the data delivery model of each flow (continuous or event-driven). STCP is an end-to-end transport protocol. A session establishment is required prior to sending any data. The session initiation packet sent by the source node informs the base station of the number of flows originating from the node, the type of data flow, transmission rate, and required reliability. Once the base station has acknowledged the session initiation packet, the node can start transmitting. Intermediate nodes trigger hop-by-hop retransmissions when the reliability level is not met. According to the characteristics of the transported data flow, the protocol detects packet loss either using positive or negative acknowledgments. In case of a continuous flow, the receiver knows when packets should arrive. Instead of sending an ACK for

each packet received, which would waste energy and bandwidth, the receiver only sends negative ACKs when a packet is not received. Since in event-driven applications, the receiver cannot predict the arrival of packets, ACK-based loss detection is used for these applications.

As shown in the above discussion of recent research work on transport protocols for WSN, we can see that efforts have been made to develop transport protocols suited to the specific constraints of WSN which support multiple types of applications and simultaneous flows. However, most protocols provide simple fairness although flows might have different priorities and bandwidth requirements. These aspects were not considered until recently. Gungor et al. proposed a real-time and reliable transport protocol for wireless sensor and actor networks (WSAN) called $(RT)^2$ [12]. This transport protocol addresses congestion control and timely event transport reliability, providing specific delay bounds support and heterogeneous reliability. Indeed, sensor-actor and actor-actor communication do not have the same reliability requirements; in the sensor-actor communication scheme, 100% reliability is not needed assuming that a given event will be detected by several sensors, hence causing data redundancy, while actor-actor communication needs 100% packet reliability to avoid inaccurate action decisions. This reliability is achieved by distinguishing the cause of packet loss (congestion and non-congestion related losses) and by using selective-acknowledgments (one SACK packet for every data packets received). The protocol provides low latency to real-time packets by using the general principle of earliest deadline first. Since the transport delay is also affected by the network load which depends on the nodes reporting frequency, this reporting frequency is continuously adapted according to the desired delay-constrained reliability and the network state (congested or not). In addition to WSAN, this protocol may also be suitable for WSN applications where critical information such as alarms is exchanged between nodes and the sink . Another transport protocol that performs priority-based rate control and congestion control was proposed in [13]. It is designed for wireless multimedia sensor networks and implements a service differentiation mechanism. Service differentiation allows to differentiate between best effort and critical traffic in order to achieve a better reliability and latency for critical traffic. The protocol implements four queues: the queue for high priority traffic uses strict priority scheduling and WRR is employed for the three other queues with different weights according to their priority. The congestion detection mechanism considers queue length, packet service time, and link-by-link loss detection. This protocol also performs congestion avoidance to prevent critical traffic losses by applying the random early detection or drop (RED) technique on non critical classes.

As the resources are very limited, WSN with high reporting frequency need transport protocols

to perform congestion control and to help honoring delay and reliability bounds of each flow. As traditional transport protocols were not suited for WSN, several solutions were proposed to provide congestion control and reliability guarantees. Efforts have been made to support heterogeneous applications and data delivery models. However, the existing transport protocols for WSN still have limitations. Cross-layer optimizations are required to reliably detect congestion, which should not be confused with packet loss due to collisions. Also, transport layer protocols should consider the type of routing protocol implemented at the network layer: single-path or multipath. Multipath routing can be an issue regarding congestion control. Nevertheless, reliability may benefit from interactions between these two layers (e.g. for efficient re-routing).

2.4.2 Network layer

During the last decade, considerable research efforts have been paid in developing energy-efficient routing techniques for WSN. These routing protocols consider energy efficiency as the main objective with the assumption that data does not have stringent QoS requirements. As a consequence, their performance is not satisfactory when they are used in demanding applications with more constrained traffic. The current research trend considers QoS constraints in order to allow real-time and bandwidth-hungry applications such as multimedia surveillance. A survey on energy-efficient routing techniques was provided by Ehsan and Hamdaoui, along with a study of routing protocols with QoS support [14]. In the following, we give an overview of research efforts towards QoS-aware routing.

The requirement of low latency communication is getting more and more important in emerging applications. Out-of-date information is irrelevant and leads to unnecessary network load. Real-time routing protocols aim to bound the end-to-end transmission time by finding the optimal path and reducing the queuing delay for real-time packets. Different methods may be employed to find the optimal path. SPEED [15] is a well-known geographic routing protocol which allows real-time communications in WSN. Geographic routing protocols try to estimate the end-to-end delay by computing the distance between the source and the sink. This distance is estimated using the location information obtained through localization techniques. SPEED collects local information about neighboring nodes by exchanging beacons and exploiting feedback information such as the elapsed time between the transmission of a packet and the reception of an ACK, or the miss ratio of the neighbors (when they could not provide the desired speed). In contrast, in tree-based routing protocols, the overall cost of the available routes is estimated in order to

find the best path. The cost of routes may be computed considering different metrics such as energy, transmission delay, link erros, available bandwidth, etc. SAR [16] is an early proposed routing protocol which considers end-to-end delay and energy efficiency. The algorithm creates multiple trees rooted from nodes situated at one hop from the sink, and multiple paths are maintained in the routing table of each node. The cost of each path is an additive metric computed considering an energy cost and delay of each link. Path selection is made by the node that generates the packets to send by choosing the path for which the level of QoS provided to each packet suits the priority level of the packet.

In addition to reducing end-to-end delay, some routing protocols also aim to improve the reliability. MMSPEED [17] is an extension of SPEED that adds a differentiated priority packet delivery mechanism and improved reliability using multipath forwarding. MMSPEED provides multiple QoS levels in the timeliness domain by using different delivery speeds (different relay nodes), while different reliability requirements are supported by probabilistic multipath forwarding. The number of delivery paths is based on the required end-to-end reaching probability. In order to reduce the queuing delay of incoming packets, they are classified into three queues according to the packet priority. The highest priority queue is served first, followed by the medium priority queue. The low priority queue is served last. In addition, in order to favor priority transmissions among neighbors, the authors indicate that a special support from the MAC layer is needed so that the channel access latency is reduced for high priority packets. Ahmed and Fisal proposed RTLD [18], a real-time geographic routing protocol for WSN which provides efficient power consumption and high packet delivery ratio through load distribution. The routing management consists of three sub functional processes: forwarding metrics calculation, forwarding mechanism, and routing problem handler. Optimal forwarding is performed using the following metrics: packet velocity, link quality, and remaining power. RTLD uses geodirectional-cast forwarding that increases the delivery ratio as it uses multiple paths. The end-to-end delay is reduced by choosing the forwarding nodes with the maximum packet velocity. The routing problem handler serves to recover from routing failures using power adaptation and feedback control packets. Similarly, Ben-Othman and Yahya proposed an energy efficient and QoS aware multipath routing protocol for WSN called EQSR [19]. EQSR uses service differentiation to allow real-time traffic to reach the sink within an acceptable delay while providing best-effort service to delay tolerant packets. It implements two queues: one for real-time traffic and the other one for non real-time traffic, and employs a strict priority scheduler to set the next packet

to transmit so that real-time traffic is favored. High reliability is provided using multipath routing and forward error correction. Before the transmission, the packet is split up into several sub-packets, error correction codes are added, then the sub-packets are sent across the available multiple paths. The paths with the lower end-to-end delay are used for RT traffic. Average end-to-end delay and packet delivery ratio are slightly affected by the increase of the node failure probability because of the error correction scheme. Indeed, to a certain extent, the original message can be reconstructed using the generated XOR-codes. The network lifetime is maximized through balancing energy consumption across multiple nodes.

Given the increasing development of multimedia applications for WSN, some protocols are proposed to handle real-time video streaming. In addition to delay, bandwidth has to be considered. Real-time multimedia streaming requirements in terms of bandwidth are considerable and may reach or even exceed the maximum transmission capacity, which is very limited. With typical WSN transceivers, the maximum data rate is 250 kbit/s. In contrast, the Wi-Fi technology allows data rates up to 54 Mbit/s, or even 150 Mbit/s for the 802.11n protocol. Multipath routing offers many advantages compared with single-path routing: it enables load balancing and allows to increase the transmission capacity. Chen et al. proposed Directional Geographical Routing (DGR) [20] which investigates H.26L real-time video communications in WSN. DGR divides a single video stream into multiple sub-streams and exploits multiple disjoints and non-interfering paths to transmit these sub-streams in parallel thus forming a larger aggregated bandwidth. Usually, reliability is of second importance since a low loss rate may be tolerated by this type of applications for uncompressed or redundant video streams. However, H.26L video data is highly compressed and extremely sensitive to transmission errors due to the frame redundancy. Most routing protocols in WSN employs multipath routing in order to improve the reliability of a single flow. DGR delivers multiple sub-streams over multiple disjoint paths and the responsibility of reliable data delivery is relieved by the use of forward error correction (FEC) coding. FEC is an error control technique used to recover from transmission errors due to the use of an unreliable communication channel. FEC allows to often correct these errors thus avoiding to retransmit the corrupted packets, which is ineffective in real-time applications where the retransmitted data would be received out of date. This protocol allows good received video quality.

Meeting QoS requirements in WSNs may introduce an overhead into routing protocols in terms of energy consumption. This overhead is unavoidable for applications which have strict delay and bandwidth requirements. Although most of the proposed protocols try to reduce energy

consumption, QoS support is a trade-off to the network lifetime. As novel QoS critical WSN applications proliferate, new challenges continually appear. Research efforts have been made to provide routing protocols supporting real-time traffic and allowing reliable data delivery for several applications including multimedia applications. Nevertheless, open issues are still to be resolved in QoS routing for WSN. Few routing protocols are designed to support heterogeneous applications. Routing algorithms should be flexible to support multiple data flows with different QoS requirements (delay, reliability constraints). Jitter has not been considered although it is a serious issue in interactive real-time applications. Dynamic and adaptive protocols are required to handle/exploit mobility. Most of the existing routing protocols do not take mobility into consideration. Supporting mobility is an interesting feature in tracking applications. It may also be used to overcome holes to improve the network lifetime when routing protocols do not find alternate paths. Finally, efficient routing solutions could emerge from a joint design with other layer protocols.

2.5 QoS provisioning at the MAC layer

In wireless networks, the MAC layer plays a key role in QoS provisioning. Since the radio channel is shared and cannot be accessed simultaneously by several nodes, the overall network performance depends directly on the optimal management of this resource. In this section, we provide to the reader some basic knowledge of radio communications issues, followed by an overview of design-drivers for WSN MAC protocols. Finally, we provide a state of the art of QoS MAC protocols, then we point out remaining research challenges and open issues.

2.5.1 Overview of medium sharing

Wireless sensor nodes use radio waves to communicate. The transmitter produces radio waves that oscillate at a particular frequency. In order to receive the communication initiated by the transmitter, the receiver must tune in to this frequency. This radio frequency defines the communication channel or medium. Common transmitters have an omnidirectional antenna and electromagnetic radiations propagates in three dimensions. Radio communications are interference prone: other signals at similar frequencies may disturb the ongoing communication, therefore only one device may transmit at a time. The medium is said to be shared.

The notion of shared medium is depicted in Fig. 2.3. This example scenario show two nodes, A and B, and a base station S. The circles show the radio coverage of each device. To ensure that two nodes can communicate, they must be within radio range of each other. Since the radio range of A, B and S overlap, they are able to communicate with each other. The overlapping areas are called collisions domains. Indeed, if two devices within radio range transmit simultaneously, both signals interfere with each other resulting in a collision.

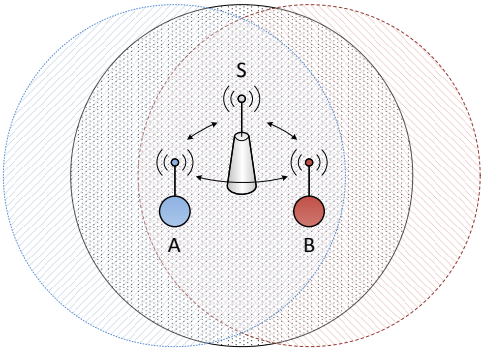


FIGURE 2.3: Illustration of shared medium and collision domain

The role of the MAC layer is to provide channel access control mechanisms to allow multiple nodes to communicate over a shared medium. We gave an overview of common channel access methods for WSN in Section 1.4.2. They fall into three categories: contention-based, contention-free, and hybrid methods. Contention-based or random access protocols exploit randomness in order to minimize the collision probability, contention-free protocols divide the available resources between contenders such that each node can use its resources exclusively without the risk of collisions, and hybrid MAC protocols combine both approaches. We notice that in contention-based an hybrid MAC protocols, collisions may happen, therefore they may implement collision avoidance mechanisms. Several collision avoidance mechanisms were proposed. A simple collision avoidance mechanism called carrier sense consists in listening to the channel before transmitting: nodes attempt to avoid collisions by transmitting only when the channel is sensed to be free or idle. In order to obtain this information, the MAC layer interacts with the physical layer through the clear channel assessment function. However, a clear channel does not guarantee the avoidance of collisions. Collisions may still happen in the presence of hidden terminals. In addition, this mechanisms may prevent feasible transmissions and lead to under-utilization of the channel. This problem is referred to as the exposed terminal problem. In what follows, we give a detailed explanation of these issues along with an overview of methods designed to overcome these limitations.

2.5.2 Shared medium issues

We identified two limitations of the carrier sense channel access method. The first one is known in the literature as the hidden terminal problem, and the second one as the exposed terminal problem. Request-to-send / Clear-to-send (RTS/CTS) is an optional mechanism implemented in the 802.11 standard to reduce collisions that partially solves both issues. We present in detail these two problems and show the benefits and the limitations of RTS/CTS.

Hidden terminal problem The hidden terminal problem or hidden node problem occurs when several nodes are visible from a sink, but are not within range of each other. An example on hidden terminals is shown in Fig. 2.4. When node A wants to transmit, it senses the channel and if it is idle, the node starts its transmission. However, sensing the channel in this situation is not relevant since A cannot hear B. Node A will find the channel free even if there is an ongoing transmission from node B, it will start to transmit anyway and the simultaneous transmissions from A and B will collide at the receiver.

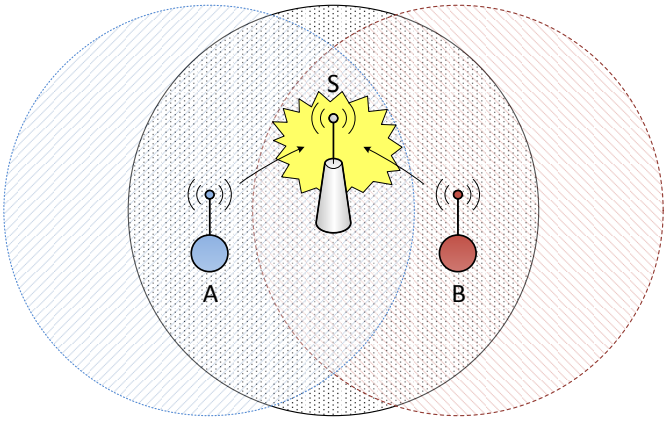


FIGURE 2.4: Illustration of the hidden terminal problem

The principle of RTS/CTS is to establish a handshake between the source and the destination prior to any data transmission. A node wishing to transmit data initiates the process by sending a request-to-send message, and the destination replies with a clear-to-send message. During this exchange, other nodes may hear the RTS or CTS frame and refrain from sending data for a given time specified in the frame. In Fig. 2.5, we illustrate how RTS/CTS solves the hidden terminal problem, i.e., how it allow to avoid collisions. Node A sends a RTS to the sink which replies with a CTS. Node B hears the CTS, thus it is informed that a transmission will occur in its collision domain and stays quiet during node A's transmission.

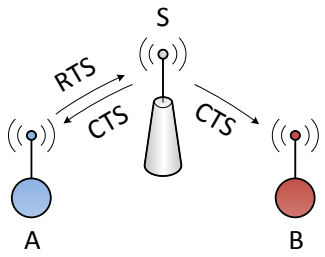


FIGURE 2.5: RTS/CTS and the hidden terminal problem

Exposed terminal problem The exposed terminal problem or exposed node problem occurs when a node is prevented from sending packets due to a neighboring transmitter although the two receivers are out of range of each other. An example is shown in Fig. 2.6. If node B senses the channel when A is currently transmitting, it will found the channel busy and delay its transmission, while it would have been possible to transmit without disturbing the other communication. This results into an under-utilization of the medium.

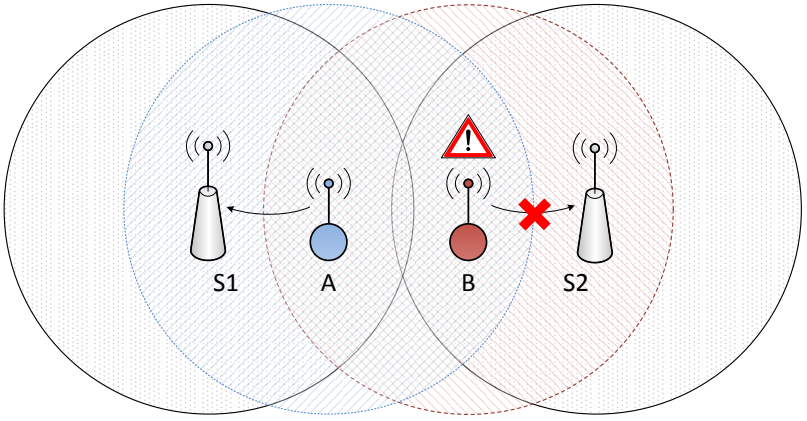


FIGURE 2.6: Illustration of the exposed terminal problem

RTS/CTS exchange allows to overcome this situation. Indeed, if node A wants to send data, it will send a RTS frame to the destination, and the sink will send back a CTS frame. Node B will hear the RTS but not the corresponding CTS, so it will deduce that it is not within range of the receiver of this communication, and that it may transmit as well.

However, the exposed node problem persists when S1 and S2 are the sources, and A and B are the destinations. We suppose that S1 initiates a transmission first: it sends a RTS frame to A which replies with a CTS. The CTS is heard by B which will enter a quiet period. Then, S2 sends a RTS to B but it will not be able to respond, thus preventing S2 from transmitting, although it would not have interfered with transmission from S1 to A.

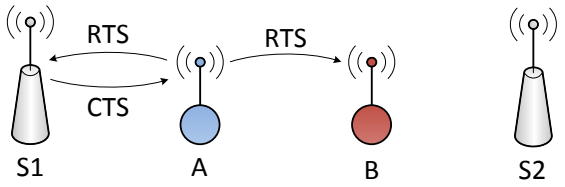


FIGURE 2.7: RTS/CTS and the exposed terminal problem

RTS/CTS was introduced in order to minimize the risk of collisions, and to solve the hidden terminal and exposed terminal problems. However, when using RTS/CTS, there are still situations where collisions occur and it may prevent potentially successful transmissions. For instance, since neighboring nodes do not always have the same transmission ranges, a CTS sent by a station may not always be heard by potential transmitter. In addition, RTS and CTS frames may also interfere with ongoing transmissions and collide, even though the probability of collision is low since RTS packets are small. Finally, RTS/CTS handshake introduces latency and overhead.

2.5.3 Design-drivers for WSN MAC protocols

The goal of an efficient MAC protocol is to provide a fair channel access to multiple sources while reducing the risk of collisions, maximizing the achievable throughput, and minimizing energy consumption. Designing an efficient MAC protocol for wireless sensor networks requires a thorough understanding of the main causes of bandwidth and energy waste. The following issues have to be considered in order to optimally make use of the limited resources of WSN.

Collisions Reducing collisions is a major concern in WSN due to its large impact on the network performance. Indeed, the energy drained in the transmission and reception of collided frames is just wasted, as well as bandwidth. In the previous section, we presented common scenarios where collisions may occur and we discussed RTS/CTS. The IEEE 802.11 standard employs the RTS/CTS mechanism to avoid collisions for large packets. However, this mechanisms may not be suited for wireless sensor networks due to their limited resources. RTS/CTS is not energy efficient as the transmission of extra packets accounts for extra energy expenditure. In addition, in many WSN applications, data packets are often as small as RTS packets. Therefore, they have identical collision probability, so it is not worth exchanging RTS/CTS messages since it adds an overhead and deteriorates the throughput instead of improving the performance. Receiver-initiated collision avoidance mechanisms are an alternative to RTS/CTS which adopts

a sender-initiated scheme. The basic principle of such schemes is that data exchange is initiated by the intended receiver. The receiver periodically invites the potential senders to transmit by polling neighboring nodes. This technique enables asynchronous duty cycle mechanisms which may help to save energy. It allows receivers to switch off their radio in order to avoid idle listening and save energy during sleep periods. Receivers periodically wake up to trigger the pending transmissions without a need for synchronization. Nevertheless, the use of this mechanism may degrade the latency compared to that of sender-initiated protocols, as queuing delays increase along the duration of the sleep period. Reservation-based protocols such as TDMA avoid collision problem by finding communication schedules whereby interfering nodes do not transmit at the same time, at the cost of increased latency and degraded throughput under low traffic conditions.

Overhead The overhead of a communication protocol is the ratio of resources (time, memory, energy, bandwidth) used to perform other functions than the actual protocol goal. For example, control packets used in some protocols that do not carry application data, like RTS and CTS packets, and lead to the consumption of extra bandwidth thus reducing the channel capacity for actual data transmission. The overhead of the RTS/CTS exchange is illustrated in Fig 2.8. For a transmission of a data packet whose size is comparable to that of RTS and CTS messages, the extra time consumed by the RTS/CTS handshake is approximately equal to the time spent for transmitting the data packet and receiving an ACK. In this example scenario, the exchange of RTS/CTS induces an overhead of about 50%. Sleeping time and backoff timers also consume time and result in a reduced throughput as well. Nevertheless, some overhead is unavoidable in order to efficiently organize and control the transmission of nodes within a collision domain, and to improve QoS metrics such as latency.

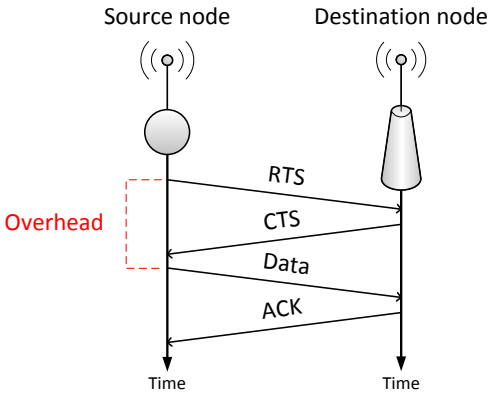


FIGURE 2.8: Overhead of RTS/CTS exchange

Overhearing and idle listening Energy efficiency is a major concern in designing WSN communication protocols, since the network must keep functioning as long as required by the application. In most sensor network applications, the energy consumption is dominated by the node's radio consumption. Since the radio is controlled at the MAC layer, MAC protocols are a key stone in optimizing the lifetime of the network. Special attention must be paid to the use of the radio, in particular to avoid unnecessary transmissions and receptions. For instance, energy may be drained when listening to irrelevant transmissions (packets intended for other nodes, redundant broadcast, etc.). This issue is called overhearing. Another situation where energy is wasted is idle listening, when nodes keep their radio on and listens to the channel, waiting for potential incoming packets.Indeed, the amount of energy consumed by the radio in the idle state, although significantly lower than that of receive and transmit mode, is far from negligible, as shown in Table 2.1 (these values were extracted from the datasheet of the CC2420 transceiver [21]). Therefore, energy-efficient MAC protocols should make nodes sleep during long periods of inactivity. Nevertheless, sleeping receivers should not miss incoming packets. Coordinating transmissions and receptions when duty-cycling is implemented is a challenging issue.

Mode	Energy consumption (mA)
Receive mode	0.426
Transmit mode at -25 dBm	8.5
Transmit mode at 0 dBm	17.4
Idle mode	18.8

TABLE 2.1: Energy consumption of a typical radio (CC2420)

2.5.4 Design approaches for QoS provisioning at the MAC layer

MAC protocols are responsible for the difficult and critical task of organizing the channel access and achieving an appropriate performance according to the application requirements. Due to the limited resources of WSN, QoS-aware MAC protocols are required to allow the development of WSN applications with high requirements. The performance of MAC protocols can be expressed in terms of energy efficiency, throughput, latency, reliability, jitter, and fairness. In the following paragraphs, we discuss how the MAC layer design affects these performance parameters.

Energy The MAC layer controls the radio, which is the more power consuming unit in a sensor node. As a consequence, significant energy savings may be achieved only with an efficient MAC protocol which employs the radio parsimoniously. Since data transmission and reception are highly power consuming, collisions should be avoided, so there would be no need to retransmit lost packets and no energy is wasted. The use of control packets should be reduced. In order to avoid overhearing and idle listening, nodes may periodically shut down the radio. Choosing a low duty-cycle allows high energy conservation since the radio is off most of the time, however the reception of traffic concentrates on a small time window thus leading to high competition between senders. Long sleep periods also induces a significant latency, particularly in multihop networks. At the opposite, very short sleep phases are not worth it due to start-up costs which may exceed the energy saved when the radio was switched off.

Throughput The global channel utilization is largely determined by the employed channel access strategy. Contention-based MAC protocols achieve high channel utilization under low contention, however as the traffic load increases, the collision probability rises leading to high packet loss and inefficient use of the channel. Contention-free protocols achieve high channel utilization under high traffic conditions but cause increased latency when the contention is low, since nodes have to wait for their reserved time slot to transmit. Hybrid approaches try to combine contention-based and contention-free approaches in order to handle variable traffic loads, sometimes at the cost of a greater protocol overhead, which may consume extra bandwidth. The channel access method also determines the relative throughput of each node: more bandwidth may be allocated to priority nodes, for example by reserving more time slots for these nodes or by increasing their channel access probability. In the presence of several types of traffic, traffic scheduling mechanisms allow to control the relative throughput of each traffic class within the nodes.

Latency At the MAC layer, the latency is caused by queuing and channel access delays. In order to support real-time communications, an efficient MAC layer should implement QoS mechanisms that aim to minimize the latency for high priority packets. Scheduling algorithms are able to provide low queuing delays to priority traffic by delaying other kinds. In addition, the channel access algorithm should be designed so that nodes with high priority packets benefit from low channel access delay that nodes with less important traffic.

Reliability Packet loss may degrade the achieved reliability. In wireless networks, the main reasons for packet loss are transmission errors and collisions. In order to provide high reliability, MAC protocols have to implement exclusive channel access or efficient collision avoidance techniques. Since packet loss still happens, retransmission is required to recover the lost data in critical applications.

Jitter Jitter is a serious issue for interactive real-time applications like voice call. When the jitter is high, packets may not be received in order. Packets reordering may necessitate the use of a buffer at the receiver, which may cause a detectable delay before the start of the playback. Usually in such applications, out-of-order packets are just dropped so there is no additional delay in the conversation, at the cost of degraded voice quality. Although jitter is mostly caused by routing, especially by multipath routing protocols, MAC layer protocols for jitter-sensitive applications should guarantee constant queuing and channel access delays.

Fairness Fairness is a metric that reflects whether resources are fairly shared among users. There are several definitions of fairness. Depending on the adopted perspective, a fair MAC layer may equally share the bandwidth among all active nodes, or in proportion to the amount/type of traffic of each node. The most important concern is that no user experiences starvation.

Many WSN applications combine different requirements (e.g., low latency and high reliability), and the main issue in the design of communication protocols is to optimize these metrics in order to maximize the network performance . This is particularly challenging at the MAC layer, responsible for the data transmission, which is also the most power consuming function in sensor nodes. Fig.2.9 illustrates the interdependence of design factors and the complexity of finding satisfactory compromises between competing goals. For instance, a MAC protocol which was designed to provide high reliability may achieve poor throughput and high energy consumption due to the overhead incurred by control messages, acknowledgments, and retransmissions. Efficient MAC protocols need to trade several design goals such as energy, reliability, throughput, etc. while considering application constraints (network size, type of traffic, etc.). As the design of MAC protocols is driven by constraints and QoS requirements of applications, designing efficient solutions for demanding and heterogeneous applications is challenging in comparison to single-purpose applications. In the following, we survey recent research efforts on QoS-aware MAC protocols for WSN in order to evaluate the advances in this area and point out remaining research issues.

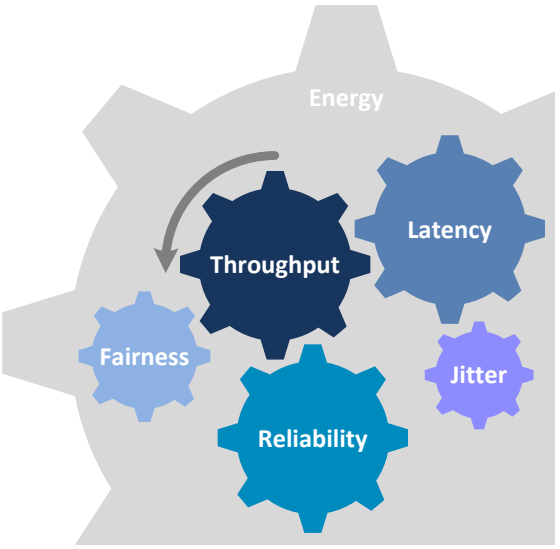


FIGURE 2.9: Interdependence of design factors

2.5.5 State of the art of QoS-aware MAC protocols for WSN

Due to the unique resource constraints and application requirements of sensor networks, existing MAC protocols cannot be used, as discussed in [1]. Therefore, since the beginning of research in the field of wireless sensor networks, many MAC protocols we proposed to tackle the limitations of these networks. Initially, the main design goal of MAC protocols for WSN was to maximize the network lifetime. The multiple surveys and classifications of MAC protocols for wireless sensor networks account for the important research work [22–24]. As there is a wide variety of envisioned applications whose requirements are very different from one application to one another, no single MAC protocol can fit all applications and each application may require a different solution at the link layer. Several application-specific characteristics such as interactivity and reliability influence the network design. Thus, the underlying network must provide guarantees in terms of latency, bandwidth and packet loss, just to name a few. There are many application-specific MAC protocols in the literature (e.g., delay-sensitive applications, bandwidth-hungry, mission-critical, etc.) as shown by the surveys provided by Teng, Suriyachai and Ullah [25–27], among others. However, there are still few QoS-aware MAC protocols, i.e., protocols which aim to accommodate different types of QoS-constrained traffic and to adapt to variable traffic loads, though QoS provisioning and service differentiation are needed in order to deliver heterogeneous traffic. In the following, we survey recent research efforts in providing MAC protocols with QoS support for heterogeneous applications while discussing their strengths and weaknesses.

Contention-based protocols Saxena et al. [28] proposed a QoS MAC protocol for wireless multimedia sensor networks (WMSN), as multimedia applications commonly carry heterogeneous traffic with different QoS requirements. This protocol is based on a CSMA/CA approach and attempts to fulfill end-to-end delay and bandwidth requirements of three types of traffic (streaming video, real-time and best-effort) using an adaptive contention window (CW) and a dynamic duty cycle for energy conservation. Service differentiation is achieved using multiple queues and a value of CW related to the traffic priority. Traffic of utmost importance will be assigned a small contention window to have a better chance of accessing the media. CW size and duty cycle are adjusted according to network statistics as transmission failures and dominant traffic type. A similar idea is pursued in the work of Yigitel et al. [29] which proposed a comparable protocol named Diff-MAC. Diff-MAC uses a different approach for intra-node packet prioritization and CW size adaptation. Saxena et al. MAC implements one FIFO queue per class of traffic before the packets are scheduled for sending, whereas Diff-MAC provides a fair prioritization of packets within the same class based on the hop count metric of each packet and uses a weighted fair queuing (WFQ) method to control the relative throughput of each traffic class. Also, Diff-MAC continuously adapts the CW size while Saxena et al.'s MAC waits for the neighboring nodes to adjust it, so in Diff-MAC CW converges more quickly to its optimal size. These two protocols use similar mechanisms to the IEEE 802.11e standard,particularly with respect to medium access prioritization. The hybrid coordination function (HCF) includes a method of channel access called Enhanced Distributed Channel Access (EDCA). EDCA defines four priority classes called access categories (AC) : Background, Best-Effort, Video and Voice. The priorities are implemented using contention windows : Voice and Video have smaller contention windows than Background and Best-Effort traffic in order to maximize the chance to transmit the priority traffic before delay-tolerant traffic. Saxena et al.'s MAC and Diff-MAC provide significant improvements over classic CSMA/CA approaches: they exhibit better performance in terms of throughput and latency. Diff-MAC also achieves fairness among the different traffic classes. However, although dynamic mechanisms enable the network to accommodate time-varying traffic loads, they introduce a significant complexity. Besides, contention-based protocols may not be efficient under high contention as RTS/CTS exchanges consume extra bandwidth. This overhead does not allow to reach an optimal channel utilization.

Contention-free protocols Contention-free MAC protocols like TDMA perform well under heavy traffic loads since scheduled transmissions allow avoiding collisions. Nevertheless, under

low contention, TDMA leads to low channel utilization and high latency. In addition, TDMA requires synchronization and is less scalable than contention-based protocols. Therefore, pure TDMA approaches are not suitable for WSN, in particular under variable traffic loads.

Hybrid protocols The limits of contention-based and contention-free MAC protocols have led to the development of hybrid MAC protocols which attempt to combine the advantages of both approaches. Z-MAC [30] protocol proposed by Rhee et al. is based on this paradigm: it dynamically adjusts its behavior between CSMA and TDMA depending on the level of contention in the network. During the setup phase, the nodes run the following operations: neighbor discovery, slot assignment, local frame exchange, and global time synchronization. The two-hop neighbor list is used as an input to the time slot assignment algorithm called DRAND [31]. This algorithm computes a schedule where two nodes within a two-hop communication neighborhood cannot be assigned to the same slot. When the setup phase is over, the transmission phase begins. Nodes can transmit during their own time slot but they may also contend to steal a slot that is not used by its owner so the channel utilization is optimized. Before transmitting, nodes take a random backoff within a given contention window. When the backoff expires, they run a clear channel assessment (CCA) to know if the channel is clear. The CW size is set in such a way that owners are always given a better chance of accessing the channel. This mechanism makes Z-MAC robust to synchronization errors. In case of clock drift, the performance of Z-MAC is similar to that of CSMA. To overcome the high overhead of RTS-CTS, this mechanism is not used in Z-MAC. Instead, Z-MAC implements two modes of operation: low contention level (LCL) and high contention level (HCL). When high contention is experienced, an explicit contention notification is sent causing the nodes to switch to HCL mode where nodes are no longer allowed to steal slots owned by two-hop neighbors. Z-MAC dynamically adjusts its behavior depending on the level of contention in the network, thus achieving high channel utilization. However Z-MAC is not suited for heterogeneous applications since it does not implement any service differentiation mechanism and QoS provisioning. I-MAC [32] adds a prioritization scheme to Z-MAC and aims to take into account the traffic load for each sensor node according to its role in the network. Higher priority will be assigned to nodes having a lot of packets to send, such as cluster heads, allowing these nodes to have a better chance to access the medium than their low-priority neighbors. Four priority levels are implemented using custom CW sizes for each priority group. Although I-MAC achieves a slightly better channel utilization than Z-MAC, it has not been designed to support QoS-constrained traffic either. Moreover, I-MAC may be hard

to deploy on a large number of nodes. Nodes are assigned a fixed priority according to their role in the network, so this implies that nodes must be manually configured, unless they are able to guess their role in the network. In addition, nodes cannot adapt their priority level in case of variable traffic conditions.

2.6 Remaining challenges and open issues

In this chapter, we have surveyed research efforts in QoS provisioning for WSN, in particular on the design of QoS MAC protocols. One of the main issue that the MAC layer has to deal with is energy consumption. The way the radio is controlled largely determines the network lifetime. Duty-cycling is a fundament mechanism in energy saving, however finding the best wake-up schedule is a complex task. Receiver-initiated schemes allow asynchronous sleep periods by shifting the responsibility of establishing communication from the sender side to the receiver side, but idle listening is also moved from receiver to senders. In WSN applications with high requirements, the way of dealing with energy efficiency has changed, since QoS has become a goal of utmost importance. QoS provisioning is a trade-off to energy efficiency: QoS support is provided at the cost of a reduced network lifetime. Besides, energy harvesting is a promising opportunity to extend the network lifetime [33]. The choice of an appropriate MAC protocol is application dependent. Contention-free and hybrid protocols provide higher throughput than contention-based protocols, but they are not scalable and require the implementation of lightweight distributed synchronization techniques as well as low complexity auto-organization/configuration and self-healing algorithms. These promising protocols have to evolve towards traffic-adaptive schemes with dynamic slot assignment techniques with low overhead that fit all traffic conditions. Multi-channel MAC protocols were recently proposed to further improve network throughput[34]. Instead of low-cost transceivers that can only operate on a single channel, they use upscale transceivers that can switch the operating frequency dynamically, thus allowing more simultaneous transmissions. Designing efficient dynamic channel allocation algorithms that adapt to the traffic conditions is a challenging problem. Most of the proposed MAC protocols do not consider variable or heterogeneous traffic. As a consequence, they perform poorly under such scenarios. Many emerging applications are multi-purpose and need that each data flow has its QoS requirements fulfilled. They require the design of MAC protocols that provide an appropriate performance for all traffic types and use the channel efficiently .

Chapter 3

Efficient QoS Provisioning at the MAC Layer in Heterogeneous Wireless Sensor Networks

This chapter deals with QoS provisioning at the MAC layer in WSN with heterogeneous traffic. Our analysis of the state of the art in Chapter 2 revealed a lack of contributions in this area. In order to fill this gap, we propose a new efficient MAC protocol with QoS support called AMPH (Adaptive MAC Protocol for Heterogeneous WSN). AMPH employs a new hybrid technique which combines time-division and random channel access in order to provide different QoS levels and achieve high channel utilization. In this chapter, we describe in detail the design and operation of our solution.

3.1 Motivation

Wireless sensor networks have the potential to revolutionize traditional monitoring tasks. Recently, new applications emerged such as target tracking, health care and multimedia applications. These complex applications often have heterogeneous sensing capabilities and require that the network supports different types of QoS-constrained traffic at variable rates. However, existing protocols are not suited for these high demanding heterogeneous applications. As the design of the MAC layer largely determines the global performance of the network in WSN, we focus on the design of an efficient QoS MAC protocol for this category of applications. This is a first step towards a full communication architecture enabling the development of these promising applications.

3.2 Design goals and assumptions

Our contribution aims to provide an efficient MAC protocol for WSN heterogeneous applications. We call these applications "heterogeneous" because of the heterogeneous nature of the traffic generated by nodes with various sensing capabilities. Indeed, the different sensors produce data that have specific characteristics, e.g., specific data rate and size, and different QoS requirements in terms of latency, reliability and bandwidth. In addition, the traffic load in these networks may be variable (from low data rates to very high data rates) and not distributed evenly among all nodes.

Typically, our protocol targets multimedia applications. In multimedia streaming applications, two types of traffic are involved: the multimedia traffic and the background traffic coming from various scalar sensors such as light and temperature. The amount of multimedia traffic is significantly higher than that of background traffic. Multimedia streams have high latency and bandwidth requirements.

We designed our MAC protocol so that is suits the distinctive features of heterogeneous traffic:

- Our protocol is able to differentiate between high priority and low priority traffic and to provide low latency and high throughput to high priority traffic.
- Our protocol is adaptive in order to operate efficiently under variable traffic loads.

Furthermore, we made the following assumptions:

- We suppose that sensor nodes are equipped with low-cost single-channel radio transceivers so that the global cost of a sensor nodes remains low.
- Since our target applications necessitate, for the most part, a relatively small number of nodes, we assume a network size from small to moderate (< 100 nodes) and a medium density.
- QoS mechanisms often involve extra control messages or in-network processing. As a result, there is a trade-off between QoS support and energy conservation. Since providing efficient QoS support is our main objective, we consider energy conservation as an objective of secondary importance.

3.3 Basic Principles of AMPH

In this section, we present in detail the design of AMPH, our new adaptive MAC protocol for heterogeneous wireless sensor networks. The basic idea of our solution is similar to that of Z-MAC: we adopt a hybrid behavior which combines the strengths of both contention-based and schedule-based approaches to maximize the channel utilization. Our hybrid channel access method allows slot-stealing thus achieving high channel utilization and providing adaptability to variable traffic loads. We also introduce a new prioritization scheme designed to fulfill the requirements of real-time traffic.

3.3.1 Hybrid time structure

AMPH uses a hybrid channel access method which combines the strengths of both contention-free and contention-based approaches. We based our hybrid mechanism on time division but nodes may transmit during any time slot using random access based slot stealing in order to maximize the channel utilization and minimize the latency. In what follows, we present the resulting hybrid time structure.

Time is divided into several recurrent time slots of fixed length. Nodes are assigned to time slots in such a way that no two nodes within a two-hop communication neighborhood are assigned to the same slot. We call nodes assigned to a given slot *owners*, the other nodes are denoted as *non owners*. More details about slot assignment are given in subsection 3.4.1. A cycle of N time slots constitutes a time frame, where N is the maximum number of time slots, which is the maximum number of contenders in a two-hop area. Fig. 3.1 illustrates the time structure where the maximum number of time slots is 3.

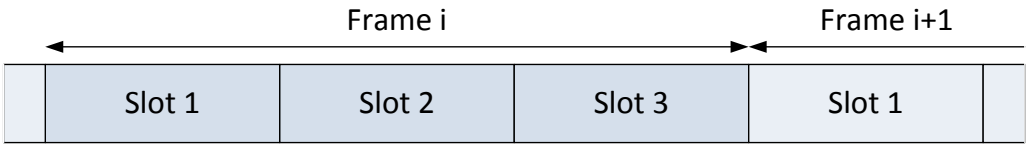


FIGURE 3.1: Time structure

Nodes having traffic to send all start the transmission process at the beginning of a new time slot (at time slot boundaries) and compete to gain access to the channel during a contention period using a random access based slot stealing scheme explained in details in 3.3.2. The winner of the competition gains exclusive access to the channel during the entire time slot. At the next time slot boundary, another competition is held and so on. We depict the structure of a time slot in 3.2.

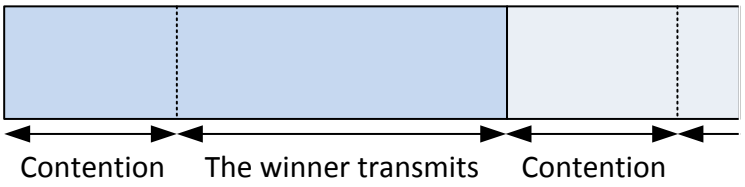


FIGURE 3.2: Structure of a time slot

3.3.2 Service differentiation and packet prioritization

We propose a new prioritization scheme in order to perform inter-nodes arbitration which ensures that nodes with high priority traffic will be able to transmit ahead of low priority nodes in case of competition to access the channel. We also define an intra-node arbitration mechanism so priority packets take precedence over other packets as soon as they are created/received. We first introduce the intra-node arbitration mechanism, and inter-nodes arbitration will be detailed subsequently.

Service differentiation and scheduling Heterogeneous WSN have to handle various traffic types with different QoS requirements. According to the assumptions formulated in 3.2 and in order to keep the operation of AMPH simple, we only consider two traffic classes: real-time (RT) and best-effort (BE). Real-time traffic takes precedence over delay-tolerant best-effort traffic so that the latency of RT traffic is minimized. We did not consider as necessary to support an intermediary class of traffic, as most of the time applications only have BE and RT traffic and no "low-delay but not-so-urgent" traffic. It would have added significant complexity.

AMPH maintains two FIFO queues corresponding to the two classes of traffic, as shown in Fig. 3.3. We assume that the traffic class is statically set at the application level by tagging the packet in a specific field of the packet header. When a packet is submitted to the data link layer

from the upper layer, a classifier checks whether the packet belongs to the RT or BE traffic class and puts it into the appropriate packet service queue. We use a strict priority scheduler to set the next packet to send, so that RT traffic always has priority over BE traffic. Our scheduler systematically selects RT packets as long as the queue is not empty, then it continues with BE packets.

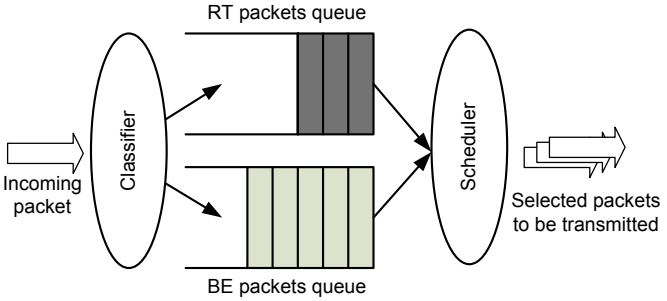


FIGURE 3.3: AMPH intra-node arbitration scheme

Medium access priority The scheduling mechanism presented above allows to select RT packets for transmission ahead of delay tolerant BE packets. An additional mechanism is needed to organize channel access between competing nodes in order to guarantee that a node having RT traffic to send has higher chance to gain access to the medium than a node having BE traffic, thus ensuring that RT traffic queuing time is minimized. We propose a new arbitration scheme that provides low channel access delay for RT packets and fairness among nodes with traffic of the same class.

In accordance with our time structure, at the beginning of a time slot, all nodes having packets to send compete to access the channel. As the medium is shared, only one node may transmit at a time, otherwise concurrent transmissions would interfere with each other. Arbitration between the nodes competing to transmit during the same slot is performed as follows. We designed our arbitration scheme using random timers called *backoff timers*, or *backoffs*. Competing nodes pick a backoff value and have to wait for the backoff duration before trying to transmit. When the backoff timer of a node expires, it senses the medium by calling the CCA function of the PHY. If the PHY returns that the channel is idle, the node may start to send packets, otherwise it has to delay its transmission. As a result, the node that obtains the smallest backoff wins the contention and gains access to the medium. When the backoff of the other contenders expires, the channel will not be idle anymore, since the winner is currently transmitting, and they will back off.

According to our design goals, RT traffic takes precedence over BE traffic, so nodes having RT packets to send should be able to access the channel ahead of nodes having BE traffic. In order to allow this behavior, nodes having RT traffic benefit from smaller backoffs that nodes with BE traffic. The contention window also depends on the role of the node: owner or non owner. Owners have priority over non owners. Since all nodes own a time slot, this system achieves a fair access to the channel among nodes having traffic of the same class. In addition, our mechanism allows non owners to steal the slots of owners when they have nothing to send, thus reducing channel access time and increasing channel utilization.

Nodes having data to send pick the backoff value β in the appropriate contention window, according to the type of traffic selected by the scheduler and if they are owner or non owner. The contention windows are non-overlapping interval set as depicted in Table 3.1. Since the backoff is chosen randomly, the probability that contenders within the same circumstances (non owners having traffic of the same class) get the exact same backoff is low and it is not likely that a collision occurs.

Owner + RT traffic	Interval A	$\beta \in [A_{min}, A_{max}[$
Non owner + RT traffic	Interval B	$\beta \in [B_{min}, B_{max}[$
Owner + BE traffic	Interval C	$\beta \in [C_{min}, C_{max}[$
Non owner + BE traffic	Interval D	$\beta \in [D_{min}, D_{max}[$
where $A < B < C < D$.		

TABLE 3.1: Contention windows corresponding to the role of the contender and the type of traffic it has to send

In Fig. 3.4, we depict an example scenario of two competing nodes *u* and *v*, where *u* and *v* both have RT traffic to send to the base station at the beginning of slot 0. Node *u* picks a backoff β_u in the interval A since it is the owner of the slot, and *v* picks its backoff β_v in the interval B. Since $\beta_u < \beta_v$, the backoff of node *u* expires first, so it runs a clear channel assessment (CCA) to determine if the channel is clear, i.e., that no nodes are currently transmitting. Node *u* finds the channel is idle, so it starts its transmission. When the backoff of node *v* is over, *v* also runs a CCA but as the channel is not idle anymore (node *u* is currently transmitting), it cannot transmit and has to wait for the beginning of the next slot (slot 1) to retry. As node *v* is the owner of slot 1, it will benefit from a small backoff and therefore will be given the highest priority to access the channel.

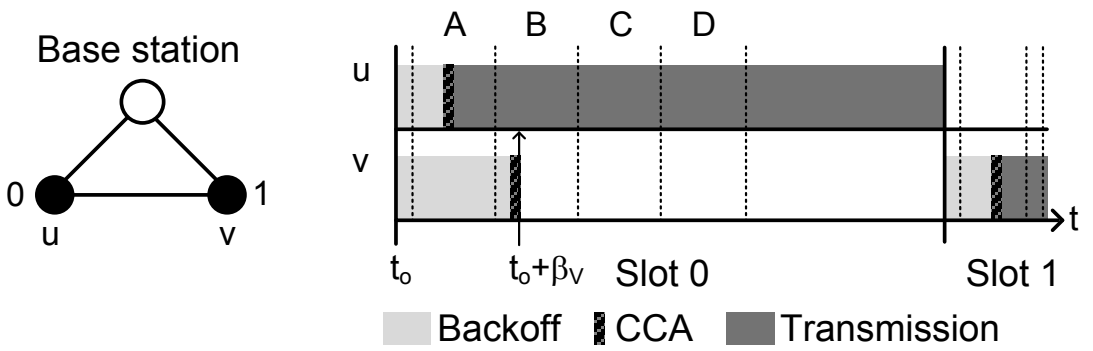
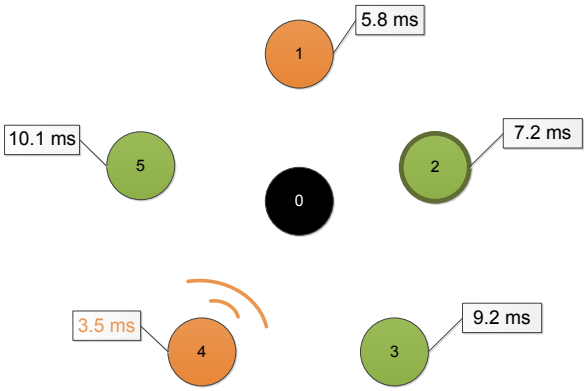


FIGURE 3.4: AMPH inter-node arbitration scheme

This example also illustrates how our backoff system ensures that AMPH is fair, i.e., that the medium is fairly shared among all nodes of the network. We can see that our arbitration mechanism guarantees that all nodes gain access to the channel at some point, in the worst case scenario, during their reserved time slot. Besides, due to the random nature of our scheme, all nodes of the same priority level have equal chance of stealing unused slots.

Now if we consider a network with more than two competing nodes, it is likely that two nodes or more will be in the same situation (non owner, same traffic type) and have to pick their backoff in the same contention window. As the backoff is chosen randomly, the probability that two nodes obtain the same backoff value is low, given that the number of contenders is small. Indeed, this probability increases as the number of competitor grows, but as stated in Section 3.2, AMPH is designed for low or medium density networks. In the following example shown in Fig. 3.5, we consider a star network of five nodes plus a base station (the base station is the black node with number 0). Nodes 1 and 4 have RT traffic in their sending queue and nodes 2, 3, and 5 only have BE traffic. We assume that node 2 is the owner of the current slot. At the beginning of the time slot, each node chooses a backoff value in the contention window corresponding to its priority level (depending of its traffic type and if its owner or non owner). Nodes 1 and 4 both are non owners of the current time slot and have RT traffic to send, so they have to choose a random backoff value in the same contention window. Node 1 got a backoff of 5.8 ms and node 4 obtained a backoff of 3.5 ms. Since node 4 had the smallest backoff, it continues the transmission process. When node 4 senses the channel, it finds the channel free since the owner of the current time slot (node 2) did not have RT traffic, therefore it starts its transmission.



	Min β_v	Max β_v
RT & Owner	0 ms	3 ms
RT & Non Owner	3 ms	6 ms
BE & Owner	6 ms	9 ms
BE & Non Owner	9 ms	12 ms

FIGURE 3.5: Illustration of chosen backoff values in a network of 5 competing nodes and the corresponding contention windows

3.3.3 Additional features

Our solution provides additional mechanisms that optimize and enhance the channel utilization and fairness.

Burst AMPH ensures that a maximum number of packets can be sent during a time slot in order to maximize the channel utilization. Indeed, transmitting a burst of packet is more efficient than transmitting only one packet per slot. It is not necessary to run the full transmission process for each packet and the overhead caused by the backoff mechanism is absorbed. The number of packets that can be sent into one burst depends on the packet size.

Starvation avoidance In our solution, we use a strict priority scheduler and a backoff mechanism which both always favor RT traffic. As a consequence, BE traffic may suffer from starvation. In order to avoid this situation, we arrange M frame among N in which BE traffic has priority over RT traffic, where N is the number of time slots in a frame and M is a parameter to adjust according to the amount of each type of traffic . During these particular time frames, the backoff values are switched so nodes having BE traffic have priority over node having RT traffic. This mechanism is optional and may be implemented only in networks with high data rate continuous RT traffic sources.

3.4 AMPH Operation

AMPH operation consists of two distinct phases, an initial setup phase, followed by the phase of normal operation called the transmission phase.

3.4.1 Setup

At startup, nodes enter a setup phase and they perform the following initialization actions: neighbor discovery, slot assignment, framing, and synchronization. Each node constitutes its two-hop neighbors list which is used as an input for the slot assignment algorithm. The slot assignment problem is analog to the graph coloring problem. In AMPH, slot assignment is performed using DRAND, an efficient distributed slot reuse scheduling algorithm also used in Z-MAC. DRAND ensures that no two nodes within a two-hop communication neighborhood are assigned to the same slot. For more details on DRAND operation, the reader may refer to [31].

In Fig. 3.6, we provide an example of slot assignment. We consider a networked formed of 8 nodes denoted A, B, C, D, W, X, Y, and Z, plus a base station S. The lines represent the connectivity links between nodes. The numbers and the different shades of gray show the slots to which nodes are assigned. Due to slot reuse, we only need 5 different slots (slot 0 to slot 4) instead of 9 (9 would have been the number of slots if we had assigned each node to a unique slot).

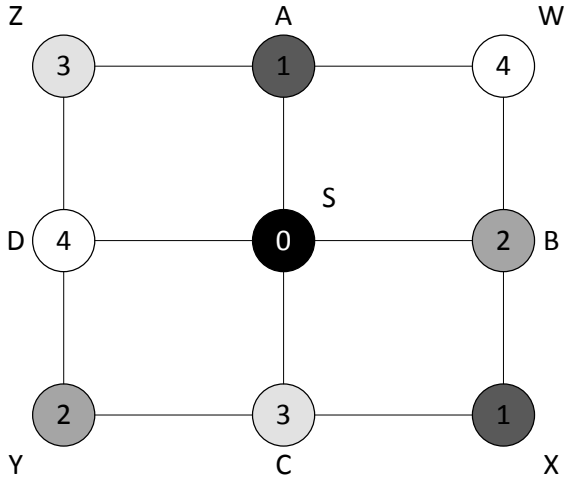


FIGURE 3.6: An example of slot assignment using slot reuse

The maximum slot number defines the time frame length, then nodes synchronize their schedule once at the beginning so they all have the same clock value and start slot 0 at the same time. When the setup phase is over, nodes begin their normal operation.

3.4.2 Transmission

As explained above, our protocol operates according to a specific time structure. The time is divided into recurrent time slots forming frames. The MAC routine occurs at the beginning of each time slot. Depending if the node has data to transmit or not, or if it receives traffic from neighboring nodes, the node performs various operations. In the following, we explain the actions performed by a node during one time slot, especially during the transmission process.

There are basically four possible scenarios:

- the node wants to transmit and the channel is idle,
- the node wants to transmit but the channel is not idle,
- the node receives data,
- the node has nothing to do.

We describe the operations of a node in these different scenarios by following the state transition diagram of AMPH given in Fig. 3.7.

Init – Wait During the setup phase, the node is in the *Init* state. After the execution of the setup, the node switches from the *Init* state to the *Wait* state. The node ends up in *Wait* at the end of each time slot and stays in this state when it has nothing to do at the beginning of a new slot. The radio may be switched off if the conditions are met: the node has no data to send, and the node is not supposed to receive any data (in a star topology for example, where every node can reach the base station directly).

Backoff At the beginning of each time slot, if the node has packets to send, it enters the *Backoff* state and performs the following tasks: it checks whether if it is the owner of the slot and if there is at least one RT packet in the queue. Next, it computes its backoff value β randomly within the corresponding window, as explained in Section 3.3. While waiting for the end of the backoff

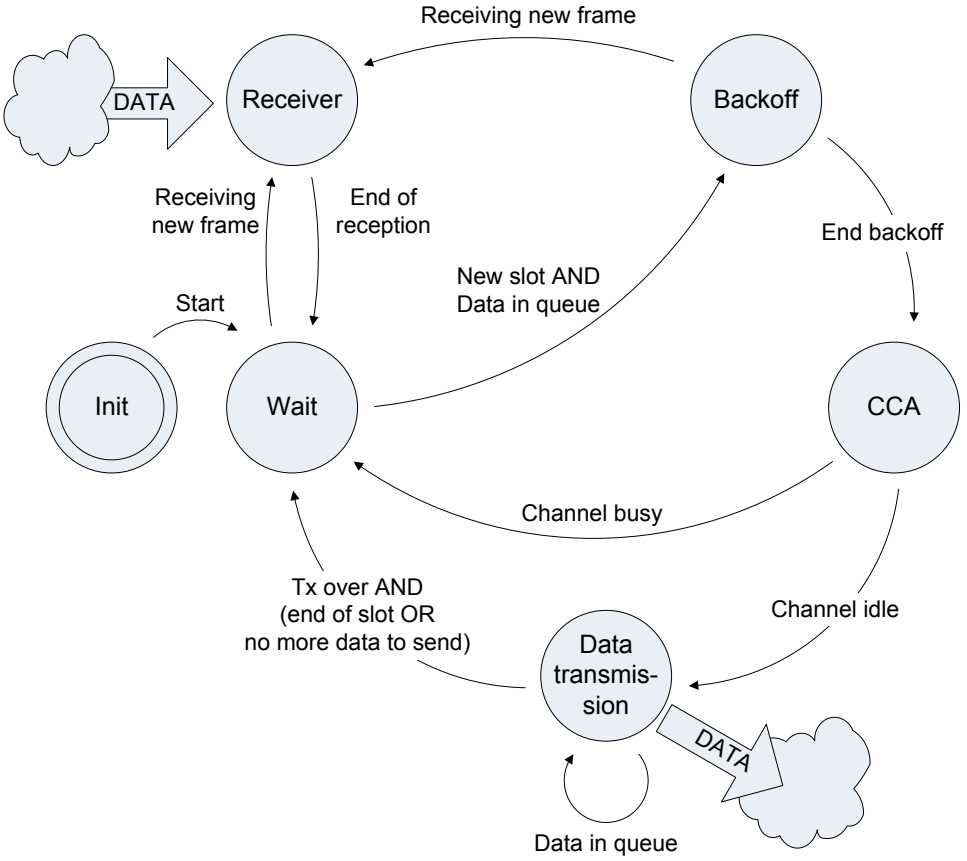


FIGURE 3.7: State machine of AMPH

time, the node stays in the *Backoff* state. During backoff, the node listens to the radio channel in the event that it receives data. If so, it becomes *Receiver*.

CCA When the backoff expires, the node switches to the *CCA* state and performs a clear channel assessment (CCA) to sense the channel. If the channel is idle, the node is allowed to begin the transmission and goes into the *Data transmission* state, otherwise it returns to the *Wait* state and waits for the beginning of the next slot to retry using the same process. As nodes listen to the radio channel during the backoff period, CCA is not necessary in this case. However, in a star topology were all nodes only communicate with the base station, the radio could then be turned off to save energy and thus CCA would be required.

Data transmission Once a node reaches the state *Data transmission*, it is allowed to transmit. The node sends packets until the queues are empty or there is not enough time left to send another packet. When the transmission is over (whatever the reason), the node returns to the

Wait state and waits for the beginning of the next slot. A similar transition to the *Wait* state happens when the node is in the *Receiver* state and the reception is completed.

Receiver In multihop networks, nodes may act as relay nodes and receive data from other nodes that need to be forwarded. Nodes have to listen for transmissions intended for them during the *Wait* and the *Backoff* states. The reception has priority over transmission. As soon as a packet is being received, the node switches to the *Receiver* state. The node leaves this state when the reception is over and returns to the *Wait* state. No other event can interrupt the reception.

3.5 Conclusion

In this chapter, we proposed a new MAC protocol with QoS support for heterogeneous wireless sensor networks called AMPH and we explained in details the design of our solution. In order to demonstrate to performance of AMPH, we conducted simulation experiments and compared the results to our best competitor in the literature. The simulation results are presented in Chapter 4. We also proposed a mathematical model of our protocol to further prove its efficiency. Chapter 5 provides detailed explanations of the modeling process along with numerical results.

Chapter 4

Performance Evaluation of AMPH through Simulation Experiments

In this chapter, we study the efficiency of AMPH through simulation experiments. We describe in detail our approach to perform this evaluation, then we analyze the relative performance of AMPH and Diff-MAC. The results demonstrate the ability of our hybrid solution to provide higher channel utilization than contention-based protocols and show that our protocol achieves a low collision rate and provides low latency for high priority traffic.

4.1 Goals

In the previous chapter, we presented in detail the design of AMPH, our new adaptive MAC protocol with QoS support for heterogeneous WSN. The goal of our solution is to provide high channel utilization, efficient prioritization of real-time traffic, and fair data delivery. In order to assess the performance of our protocol, we carried out extensive simulations and we compared the results with those of Diff-MAC, a contention-based QoS-aware MAC protocol for wireless multimedia sensor networks. We selected Diff-MAC as a competitor since it is a well-known MAC protocol for WMSN and it is the closest protocol in the literature to our protocol. Our objective is to show the benefits of our hybrid channel access technique over contention-based approaches and to demonstrate the efficiency of our prioritization scheme. In order to evaluate these abilities, we examine the following metrics: *throughput*, *latency*, and *reliability*.

4.2 Simulation environment

We implemented AMPH in OMNeT++ and we simulated the PHY using models provide in MiXiM, a modeling framework for wireless networks. OMNeT++ is a discrete event simulation environment for modeling communication networks. It provides infrastructure and tools for writing protocols and simulations. Several models exist for wired and wireless networks, mobility, peer-to-peer networks, etc. MiXiM is an OMNeT++ modeling framework which groups several models for fixed and mobile wireless networks and concentrates on the lower layers of the protocol stack. In particular, it includes the implementation of the IEEE 802.11 and 802.15.4 PHY and MAC layers. As OMNeT++ provides a component architecture for models, it is easy to reuse existing models. A high-level language (NED) allow to assemble modules into larger components and configure simulation scenarios. Modules are programmed in C++. Since this environment is free and provides building blocks for writing wireless networks simulations, we selected this solution to develop our simulations. We reused the IEEE 802.15.4 PHY module and a basic network layer module. We developed our own MAC layer and application layer. In order to compare our solution with Diff-MAC, we also implemented Diff-MAC in the simulator. Finally, we set up identical scenarios to compare the performance of AMPH and Diff-MAC.

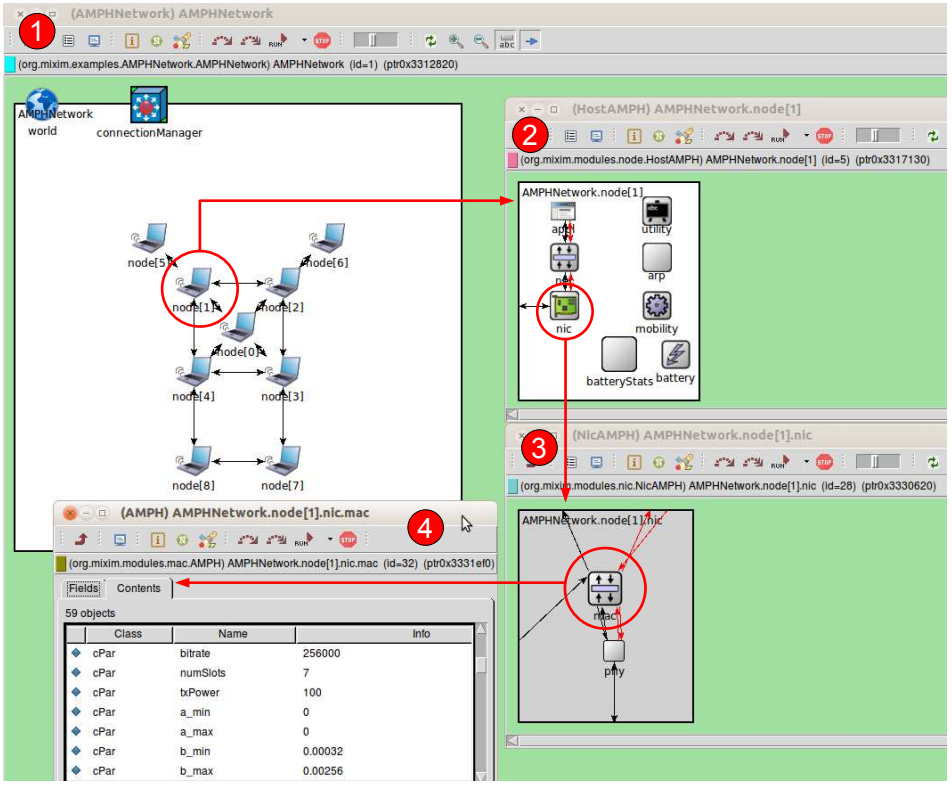


FIGURE 4.1: Modules hierarchy in OMNeT++

Fig. 4.1 illustrates the component architecture of the simulator through the inspection of the components of a node. The main window (window number one) shows the simulation of a network composed of 8 nodes and a base station (node 0). As we inspect the components of node 1 in window number two, we can see the different modules forming the communication protocols stack: Appl for the application layer, Net for the network layer, and Nic which groups the MAC and the PHY layers. Beside, we can see additional utility modules such as battery, which evaluates energy consumption. The third window shows the details of the Nic module, which contains the physical layer and the mac layer. Finally, we can observe specific parameters of the MAC layer and their value in window number four.

The integrated development environment (IDE) of OMNeT++ is based on the Eclipse platform. It provides a graphical user interface (GUI) for simulation development, execution, and results exploitation. Fig. 4.2 shows the graphical environment when running a simulation. One window draws the network topology and animated message exchanges (not visible in the screenshot). The debug information is printed into a second window. This environment is convenient to verify the protocol operation and for the debugging. The simulations may also be run using a command-line interface (CLI) which is more practical when launching several parallel simulations. Various statistics are collected during the simulations. They can be browsed directly in OMNeT++ as text or graph, and exported to several formats such as CSV and Matlab. Fig. 4.3 shows the representation of collected statistics in graph form in the OMNeT++ environment.

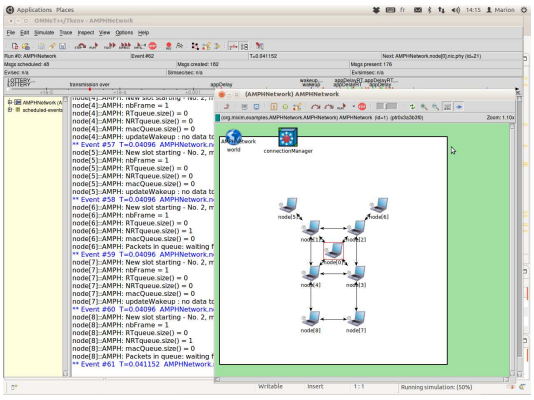


FIGURE 4.2: Running simulation graphical user interface

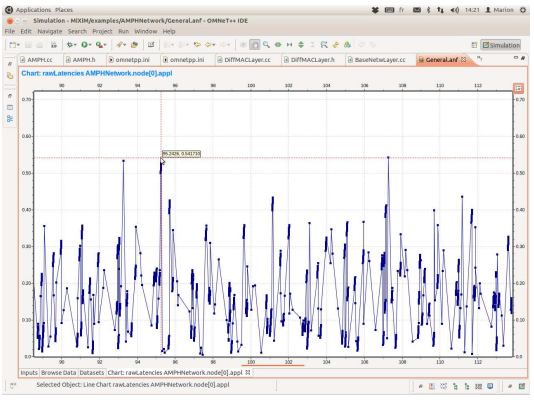


FIGURE 4.3: Simulation results

4.3 Simulation scenario and parameters

In this section, we provide a detailed overview of the implementation parameters of AMPH and of the simulation scenario. Since our protocol is designed for heterogeneous WSN with variable traffic load, we set up an example scenario similar to a multimedia monitoring application. We consider a wireless multimedia sensor network composed of nodes equipped with a video camera producing a continuous multimedia stream and with environment sensors which gather information such as temperature and luminous intensity. The application requires that the multimedia content is delivered in real-time, whereas light and temperature data are considered of secondary importance. In order to simulate this application scenario in OMNeT++, we implemented a custom application layer which generates two types of packets at different rates, corresponding to scalar data and multimedia content.

- **Simulation of scalar data.** To simulate the temperature and light measurements, our application layer generates small data packets (200 bits) whose packet inter-arrival times follow a Poisson distribution.
- **Simulation of multimedia content.** We assume that video cameras produce periodic video frames of 10 000 bits which are fragmented into 1000 bits-long packets. In order to reproduce this traffic, our application layer periodically generates 10 packets of 1000 bits.

The application layer is also responsible for setting the traffic class so that the MAC layer is able to differentiate between real-time (RT) and best-effort (BE) packets, and to provide an appropriate performance to all data flows. In our application scenario, since the multimedia traffic must be delivered in real-time and environmental data are delay-tolerant, our application layer tags video packets as RT traffic, and scalar data packets are identified as BE traffic. In order to be able to compare the performance of AMPH and Diff-MAC, the simulation of both protocols must provide the exact same traffic conditions, so we use the same application layer for the evaluation of AMPH and Diff-MAC. However, Diff-MAC implements three classes of traffic: BE, RT, and non real-time (NRT), which is an intermediary class of traffic for scalar data with higher QoS requirements than BE. In order to fit the implementation of Diff-MAC and generate NRT packets, our application layer identifies one scalar data packet out of two as a NRT packet. Since AMPH does not support this class of traffic, NRT packets are processed as BE packets.

Data generation rates are user inputs so we can evaluate the performance of AMPH and Diff-MAC under various traffic loads. The different traffic loads offered to the network to evaluate the performance are presented in Tables 4.1 and 4.2.

Mean inter-arrival time	Average created
0.1 s	10 packets/s
0.05 s	20 packets/s
0.02 s	50 packets/s
0.01 s	100 packets/s

TABLE 4.1: NRT/BE traffic loads

Frame rate
0.1 frames/s
0.05 frames/s
0.02 frames/s
0.01 frames/s

TABLE 4.2: RT traffic loads

4.3.1 MAC layer parameters

In the conducted simulations, we set the duration of a time slot such that the owner of a time slot can sent a complete video frame in one slot. Given that the size of one video frame is 10 000 bits and assuming that the available bandwidth is 256 000 bps, the duration of a time slot must be at least 39.0625 ms. We set it to 40.96 ms: it corresponds to 128 time units of 0.32 ms, which is the duration of *aUnitBackoffPeriod*, the basic time period used in the IEEE 802.15.4 MAC.

The size of the backoff intervals A, B, C, and D, expressed in time units, are provided in Table 4.3. Intervals A and C are only 1 time unit-long since there is no contention during these periods, unless the nodes are not synchronized.

Additional parameters are shown in Table 4.4.

Interval	Duration (time units)
A	1
B	8
C	1
D	8

TABLE 4.3: Backoff intervals

Parameter	Value
RT packets buffer size	50 Kbits
NRT/BE packets buffer size	4 Kbits
Available bandwidth	256 000 bps
CCA duration	0.128 ms

TABLE 4.4: Additional simulation parameters

4.3.2 Implementation of Diff-MAC

We implemented Diff-MAC according to the information provided in [29]. As Diff-MAC adopts a CSMA/CA based medium access method, we adapted the implementation of CSMA/CA provided in MiXiM to which we added the extra features of Diff-MAC: contention window size adaptation, intra-node and intra-queue prioritization. Diff-MAC uses RTS/CTS and acknowledgments. Just as AMPH, Diff-MAC sends RT packets in a burst. The length of a burst corresponds to the number of fragments of one video frame.

4.4 Simulation results

We evaluated the performance of AMPH through extensive simulations using the OMNeT++ simulation engine. We selected Diff-MAC as a competitor for our protocol and we evaluated its performance under the same simulation scenarios in order to compare the simulation results. We simulated a network of eight multimedia nodes and a base station organized in a star topology where each node is within communication range of each other and we studied the relative performance of AMPH and Diff-MAC under various traffic loads. Each scenario is simulated ten times with different seeds. In this section, we analyze the simulation results. We focus on the comparative channel utilization, average latency, and successful packet delivery ratio.

4.4.1 Channel utilization

Since high throughput is necessary for high data rate applications such as multimedia applications, achieving high channel utilization in one of the primary goals of AMPH. In Fig. 4.4, we plotted the comparative channel utilization of AMPH and Diff-MAC. As shown in this figure, AMPH achieves better throughput than Diff-MAC in all scenarios, particularly when the traffic load increases. This confirms our hypothesis that the hybrid behavior of AMPH allows high channel utilization under variable traffic loads through the use of an efficient time division schedule which enhances the contention resolution. The ability to send multiple packets in one slot also contributes in maximizing the channel utilization, as well as the fact that we do not use control messages such as RTS / CTS or ACK.

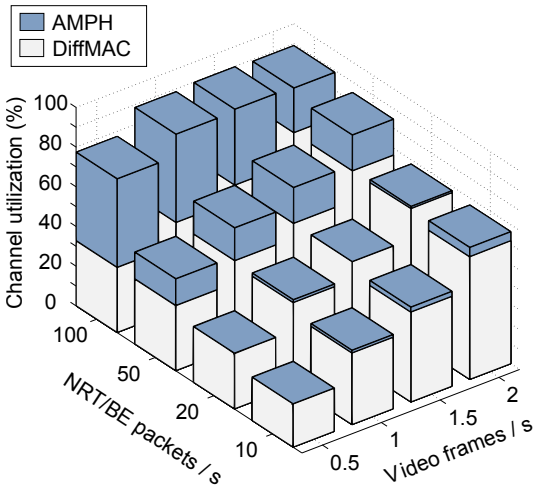


FIGURE 4.4: Comparative channel utilization

4.4.2 Latency

AMPH also aims to provide fast data delivery for real-time and mission-critical applications. In Fig. 4.5, we show the average latency of RT traffic using Diff-MAC and AMPH. At low traffic loads, the latency is very small: $\approx$ 33 ms for Diff-MAC, 45 ms for AMPH. Indeed, at low contention levels, nodes in Diff-MAC can access the medium almost immediately whereas in AMPH, the transmission process starts only at the beginning of a new slot. Nevertheless, the gap is not significant. When the traffic load increases, contention gradually increases and access to the channel becomes more difficult. Using AMPH, the latency of RT packets stays very low ($\leq$ 70 ms), thus demonstrating the efficiency of our arbitration and QoS mechanisms. At the same time, the latency of Diff-MAC rises up to 330 ms.

In Fig. 4.6, we plotted the average latency of BE traffic for Diff-MAC and AMPH. Diff-MAC supports two kinds of best-effort traffic: non real-time denoted as NRT and true best-effort denoted as BE. NRT has higher priority that BE traffic. AMPH assimilate NRT to BE traffic. In almost all scenarios, the latency of BE packets using our protocol is inferior to one second. We notice that when the BE load is set to 100 packets/s, the latency of BE packets increases up to 22 s. However, it should be noted that the mechanism that favors BE traffic over RT traffic when the BE queue fills up was not implemented. This scenario shows that even under very high traffic conditions and with no special mechanism to favor BE traffic over RT traffic, BE traffic does not suffer from starvation. Globally, we notice that AMPH behaves very well, unlike Diff-MAC whose latency rises as soon as the traffic load reaches 50% of the available bandwidth.

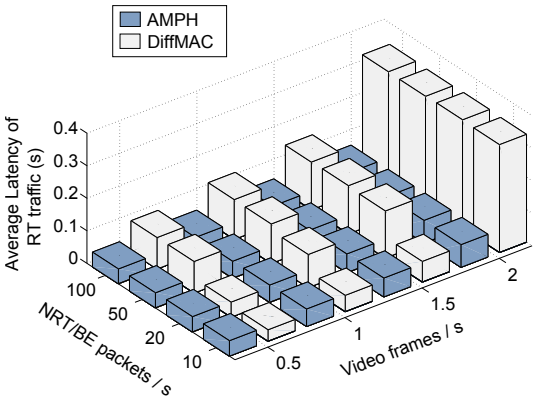


FIGURE 4.5: Comparative average latency of RT traffic

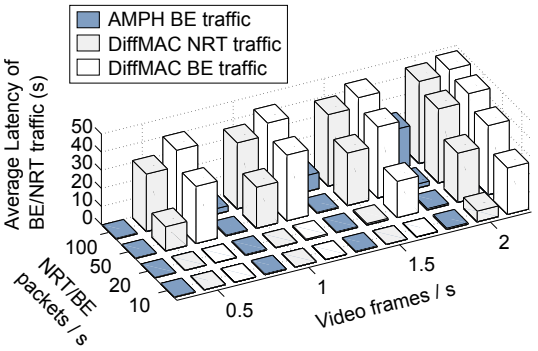


FIGURE 4.6: Comparative average latency of BE/NRT traffic

4.4.3 Data delivery ratio

Figs. 4.7 and 4.8 plot the successful packet delivery ratio performed by AMPH and Diff-MAC for all types of traffic. Reliable data delivery is an important concern, especially for critical and real-time applications, where packets loss decreases the information quality. However, for some high-throughput traffic as multimedia streaming, some packet loss can be tolerated, given that to a certain extent, coding techniques can overcome packet loss. Our simulation results show that AMPH achieves high reliability, although it does not implement RTS/CTS exchanges or packet loss recovery techniques. For real-time traffic, in the worst case scenario the reliability is 89%, and the average reliability is approximately equal to 94%, thus demonstrating that AMPH is very reliable for this class of traffic. AMPH is not only reliable for RT traffic but also for the BE traffic, since the simulation results show that the average reliability of BE traffic is also approximately equal to 94%. However, we notice that when the RT frame rate is equal to 2 frames/s and the BE traffic load is also set to the maximum, the reliability drops to approximately 50%. In this scenario, the traffic load is such that nodes encounter buffer overflows. Regarding Diff-MAC, the offered reliability for RT traffic is almost equal to one. Diff-MAC outperforms AMPH, but at the cost of poor throughput. As for NRT and BE traffic, packet loss increases as the traffic load grows. Important losses occur when the traffic load is high. The two reasons for that are that packets are dropped when they have reached the maximum number of transmission attempts and buffer overflows. According to the preferential treatment of NRT traffic over BE traffic, it suffers lower losses. Globally, we can say that AMPH outperforms Diff-MAC regarding the NRT/BE traffic, as for about half of the experiment, the reliability of Diff-MAC is lower than 50%.

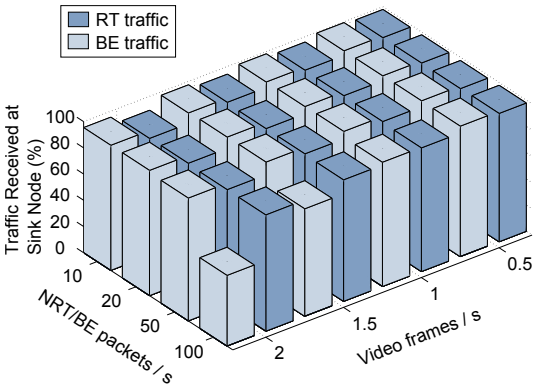


FIGURE 4.7: Comparative successful packet delivery ratio of AMPH

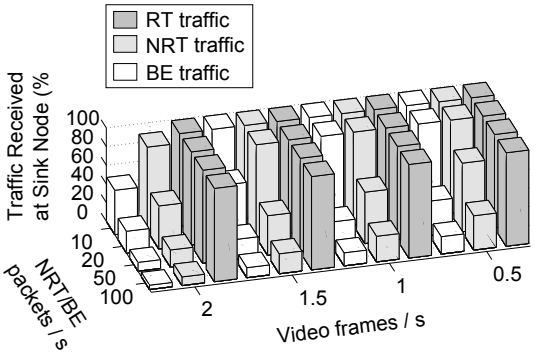


FIGURE 4.8: Comparative successful packet delivery ratio of Diff-MAC

4.4.4 Discussion / Conclusion

In this chapter, we performed extensive simulations in order to demonstrate the performance of AMPH, and compared the results with our closest competitor in the literature named Diff-MAC. The results have shown that AMPH outperforms Diff-MAC in terms of channel utilization and latency for both classes of traffic RT and BE. As for reliability, Diff-MAC offers almost a 100% reliable RT packet transmission, but at the cost of poor throughput, whereas AMPH experiences limited packet loss ($\leq 10\%$) while not wasting bandwidth with control messages. We had previously demonstrated that AMPH outperforms CSMA/CA [35]. These new experiments also tend to prove the superiority of the hybrid behavior of our approach over contention-based solutions. Our protocol effective fair service differentiation and QoS mechanisms minimize real-time traffic latency and prevent best-effort traffic starvation. The time division schedule enhances the contention resolution leading to high channel utilization and reliability. Hence, AMPH provides efficient QoS provisioning for heterogeneous traffic for a new generation of promising applications with high QoS requirements such as multimedia, tracking, and health care applications.

As a future work, we aim to demonstrate the efficiency of our adaptive behavior throughout additional scenarios under different traffic loads. In addition, we intend to further analyze the performance of AMPH through the study of energy consumption, jitter, and fairness. Finally, the protocol has to be improved in order to operate properly in multihop networks, since our slot stealing mechanism which relies on contention suffers from the hidden terminal problem. We introduce a new contention scheme in Chapter 6 in order to solve this problem and allow AMPH to support multihop communications.

Chapter 5

Analytical Performance Study

In this chapter, we provide an analytical model of our MAC protocol AMPH. The mathematical model allows the evaluation of the MAC latency by estimating the probability that a node begins a transmission within a given time and also estimates the data delivery reliability by providing the probability of success of a transmission attempt. In addition, our model shows how the network size and the distribution of traffic (proportion of RT and BE traffic) affect the performance of AMPH. In Section 5.1, we present our approach for developing the model along with some definitions and design assumptions. In Section 5.2, we explain in detail the formulation of our mathematical model, then we provide the analytical performance study of AMPH in Section 5.4.

5.1 Introduction

The design of our model follows a similar approach to that of Buratti et al. [36], where the authors provide an analytical model for evaluating the performance of the non-beacon enabled mode of the IEEE 802.15.4 standard [37]. The model provided by Buratti et al. allows the evaluation of the probability that a given sender node succeeds in accessing the channel, and that the sink receives the transmitted packet. Similarly, the goal of our model is to estimate the channel access time and the data delivery ratio of AMPH in order to perform an analytical evaluation of its performance in terms of latency and reliability. Furthermore, we aim to analyze the impact of the network size and the traffic distribution. In this section, we present some assumptions made in the model design along with the notions used in the formulation of the model, then we provide a short reminder on AMPH operation.

5.1.1 Model assumptions, reference scenario and notations

Topology We consider N nodes organized in star topology and a sink which does not transmit data. We assume that all nodes are in radio range of each other so the hidden terminal problem does not occur. Nevertheless, collisions may occur if two or more nodes sense the channel at the same time, find the channel idle and start their transmission simultaneously.

Traffic Our model is designed to allow the performance evaluation of the two types of traffic supported by AMPH: real-time (RT) and best-effort (BE).

Packet size Although AMPH may transmit several packets during one time slot, we only take into account the transmission of one packet, since it is sufficient to provide the MAC latency. As a consequence, the packet size does not affect the following results.

Resolution time In the definition of our model, the time is discrete and the resolution time is equal to *aUnitBackoffPeriod*, the base time unit in the IEEE 802.15.4. We call *aUnitBackoffPeriod* a time unit, and one time unit is equal to 0,32 ms.

The notations and symbols used in the definition of our model are summarized in Table 5.1.

Symbol	Meaning / Definition
N	Network size
$P\{T_i^j\}$	Probability to begin a transmission at the time unit j of slot i
$P\{S_i^j\}$	Probability to be sensing at the time unit j of slot i
p_s	Probability of success of a transmission
$p_{b_i}^j$	Probability to find the channel busy at the time unit j of slot i
$p_{f_i}^j$	Probability to find the channel free at the time unit j of slot i
$p_{u_i}^j$	Probability that the transmission started in (i, j) is unique
V_{S_i}	Vector containing the probability of being in each sensing state in slot i
b_v	Backoff value computed for a given node at the beginning of each slot
β_A	Upper limit of the contention window A (cf. Fig.5.4)
β_B	Upper limit of the contention window B (cf. Fig.5.4)
β_C	Upper limit of the contention window C (cf. Fig.5.4)
β_D	Upper limit of the contention window D (cf. Fig.5.4)

TABLE 5.1: Summary of notations

5.1.2 Reminder of AMPH operation

The following flow chart provided in Fig. 5.1 recalls the main steps of AMPH operation that are performed by a node having data to transmit. We highlighted the most important stages for the design of our model.

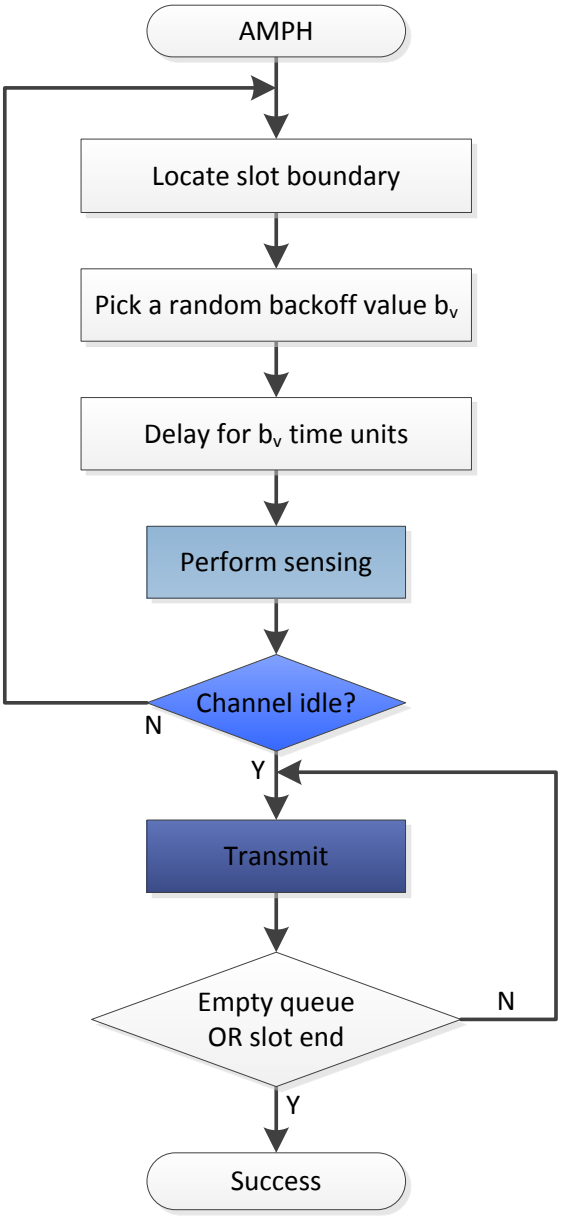


FIGURE 5.1: Flow chart representing AMPH transmission steps

5.2 Formulation of the mathematical model

The objective of our model is to provide the following metrics:

- The probability that a node begins its transmission in a given slot i at the time unit j which is denoted as $P\{T_i^j\}$
- The success probability for a transmission, i.e the probability that a node succeeds in transmitting a packet and that no collision occurs, denoted as p_s

In order to compute these metrics, we analyze in detail the transmission process of a specific node denoted as the target node. According to the operation of AMPH, a node achieving the transmission process can be in one of the four states represented in Fig. 5.2: Backoff, Sensing (S), Transmitting (T) or Idle. Idle is the default state when a node waits for the time slot boundary. At the beginning of a new time slot, a node having data to send computes a backoff value,waits for the backoff to expire, and senses the channel. After sensing, if the channel is found idle, the transmission immediately begins, otherwise the transmission is delayed and the node has to wait until the beginning of the next time slot before trying again to transmit.

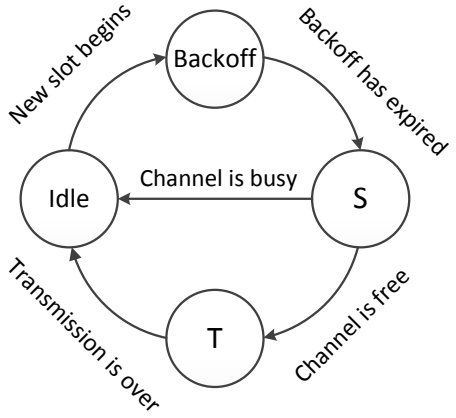


FIGURE 5.2: Full state-transitions diagram

From this analysis, we notice that the transmission of a packet is conditioned by the fact that the channel is free or busy. Evaluating the probability that the target node starts a transmission at a given time is equivalent to modeling the channel status when the node senses the channel, since we can deduce the probability that a node begins a transmission at an arbitrary time t given the probability that it was sensing the channel at $t-1$ and the probability to find the channel free at this moment.

In order to better describe the transitions between the sensing states over time and the transmitting states, we provide in Fig 5.3 the different possible sensing and transmitting states from slot 0 to a generic slot i, and the possible transitions from one state to its successors.

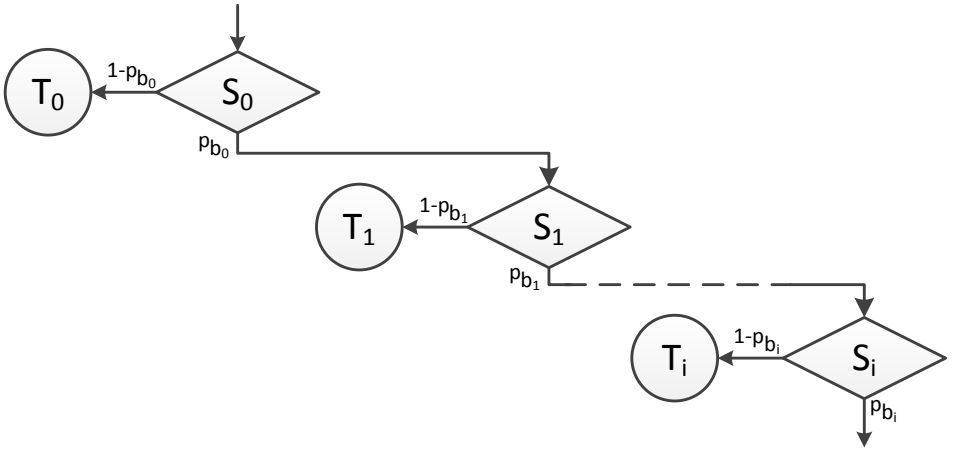


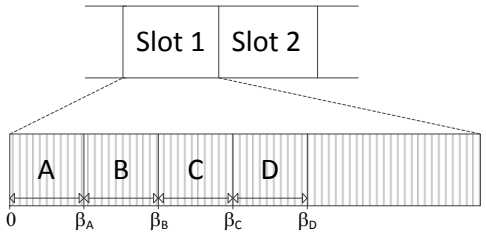
FIGURE 5.3: Representation of the transitions between Sensing and Transmitting states

A transmission may begin in slot i at the time unit j only if the channel was not busy when the sender node sensed the channel in $j-1$. Given that the probability of being in a sensing state in (i, j) is denoted as $P\{S_i^j\}$ and the probability that the channel is found busy in (i, j) is denoted as ${pb_i}^j$, the probability to begin a transmission in (i, j) denoted as $P\{T_i^j\}$ is:

$$P\{T_i^j\} = P\{S_i^{j-1}\} \cdot (1 - {pb_i}^{j-1}) \tag{5.1}$$

Since $P\{T_i^j\}$ only depends on the probability to be in the sensing state and to find the channel free, our model aims to determine all the possible sensing states and the associated probabilities to find the channel free. In the following, the sensing states are denoted as S_i^j, where i represents the slot number and j the time unit at which the node carries out the CCA function. As the CCA duration is less that one time unit, we assume that it is performed during the last $0,128ms$ of the backoff b_v, so $j = b_v$.

The backoff is modeled as follows. The backoff time value b_v is uniformly distributed in contention windows which depends on the type of traffic that the target node wants to send and if it is the owner of the current slot. The contention windows are non-overlapping intervals set as shown in Fig. 5.4.



$A = [\ 0,\ \beta_A\ [$	Owner and RT
$B = [\ \beta_A,\ \beta_B\ [$	Non Owner and RT
$C = [\ \beta_B,\ \beta_C\ [$	Owner and BE
$D = [\ \beta_C,\ \beta_D\ [$	Non Owner and BE

FIGURE 5.4: Backoff contention windows

The value of b_v can be any number between 0 and β_D, thus enabling the following sensing states: S_i^0, S_i^1, ..., $S_i^{\beta_D}$. However, the behavior of the protocol is unchanged for values of b_v belonging to the same contention window. Therefore, it is possible to group the sensing states according to the values of j: the sensing states S_i^j where $j \in A$ are grouped in the meta state S_i^A, the sensing states S_i^j where $j \in B$ are grouped in the meta state S_i^B, etc. We depict a state-transition diagram of the meta sensing states in Fig 5.5.

A node in the sensing state can become, at the next time unit, either transmitting if the channel is free, or idle if the channel is found busy (cf. Fig 5.3). If the node fails to access the channel, the node will retry to access the channel at the next time slot and compute a new backoff value according to its new role and type of traffic. The diagram represents the feasible transitions from all the possible sensing states in slot i to the possible sensing states in slot $i+1$.

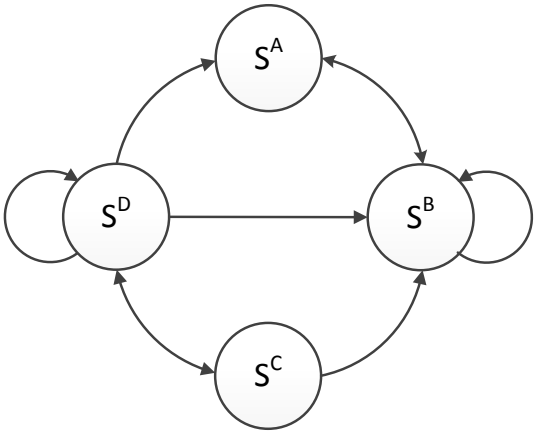


FIGURE 5.5: State-transition diagram of a generic node

The transition from state S_i^j to state $S_{i+1}^{j'}$ depends on three parameters :

- The probability to find the channel busy in $(i,\ j)$ $p_{b_i}^j$
- The role of the node in slot $i+1$
- The type of traffic the node has to transmit at the beginning of the time slot $i+1$

The transition probability from state S_i^j to state $S_{i+1}^{j'}$ depends only on the first parameter ${p_b}_i^j$, as explained below. The other two parameters determine which meta state the transition leads to. Indeed, the role of the nodes evolves and in addition, they can receive RT packets from upper layers anytime. As we want to strictly favor RT traffic over BE, if a node fails to access the channel to transmit a BE packet in slot i and receives a RT packet in the meantime, in slot $i+1$ the node will be in the sensing state S^A or S^B, while it was in S^C or S^D in i. The sending process of the BE packet is interrupted. However, in the model, we consider the process of sending a given packet from beginning to end. As a consequence, all transitions from states S^C and S^D to states S^A and S^B are impossible. We represent the remaining possible transitions in Fig. 5.6 and we further provide the associated transition probabilities, according to the role and type of traffic of the node in slot $i+1$.

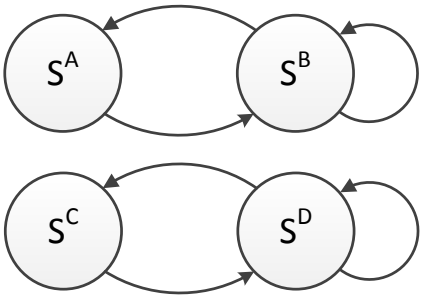


FIGURE 5.6: Simplified state-transition diagram

We denote as $P\{S^A|S^B\}$ the transition probability from state S_i^j where $j \in B$ to $S_{i+1}^{j'}$ where $j' \in A$. In Table 5.2, we give the transition probabilities of all possible transitions according to the type of traffic that the target node wants to transmit and its role at the slot $i+1$.

Node parameters	Transition probability
Node with RT traffic, owner at the slot $i+1$	$P\{S^A\|S^B\} = {p_b}_i^B$
Node with RT traffic, non owner at the slot $i+1$	$P\{S^B\|S^A\} = 0$ $P\{S^B\|S^B\} = {p_b}_i^B$
Node with BE traffic, owner at the slot $i+1$	$P\{S^C\|S^D\} = {p_b}_i^D$
Node with BE traffic, non owner at the slot $i+1$	$P\{S^D\|S^C\} = {p_b}_i^C$ $P\{S^D\|S^D\} = {p_b}_i^D$

TABLE 5.2: Transition probabilities

5.3 Calculation

In the previous section, we have formulated the basis of the mathematical model. In the present section, we explain in detail the calculation of the various elements provided during the model definition: the probability that the target node is sensing, the probability to find the channel busy, and the probability that the transmission starts and is successful.

5.3.1 Calculation of the probability to end up sensing at the next slot

Let V_{S_i} be a vector formed of the probability that the target node is in one of the four meta sensing states at time slot i.

$$V_{S_i} = \left\{ P\{S_i^A\}, P\{S_i^B\}, P\{S_i^C\}, P\{S_i^D\} \right\} \tag{5.2}$$

The probability $V_{S_{i+1}}$ that the target node end up in the four sensing states at time slot $i+1$ is:

$$V_{S_{i+1}} = V_{S_i} \cdot Trans \tag{5.3}$$

where $Trans$ is a state-transition matrix. The process is a chain, however it is not a Markov chain since our process is not memoryless. Indeed, $Trans$ depends on the history of the node, as we explain herein after.

The possible transitions from S_i^j to $S_{i+1}^{j'}$ are determined by the role of the node in the slot $i+1$ (owner or non owner), but if the node has already been owner in the current frame, it cannot be owner anymore in this frame, therefore states S^A and S^C are no longer accessible. In order to reflect this evolution of the role of the node, we represent the transition probabilities as three distinct transition matrices: $Trans_1$, $Trans_2$, and $Trans_3$. The computation of $V_{S_{i+1}}$ through Equation 5.3 uses one of these three transition matrices depending on the following scenarios:

- $Trans_1$ is used when the target node has not been owner yet and is not the owner of the next slot
- $Trans_2$ is used when the target node is the owner of the next slot
- $Trans_3$ is used when the target node has already been owner in the current frame

According to Table 5.2 provided in the previous section, the content of the matrices is:

$$Trans_1 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & p_{b_i}^B & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & p_{b_i}^D \end{pmatrix} \quad (5.4) \qquad Trans_2 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ p_{b_i}^B & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & p_{b_i}^D & 0 \end{pmatrix} \quad (5.5)$$

$$Trans_3 = \begin{pmatrix} 0 & p_{b_i}^A & 0 & 0 \\ 0 & p_{b_i}^B & 0 & 0 \\ 0 & 0 & 0 & p_{b_i}^C \\ 0 & 0 & 0 & p_{b_i}^D \end{pmatrix} \quad (5.6)$$

The probability $P\{S_{i+1}^{j'}\}$ that the target node fails to access the channel in (i, j) and ends up in the sensing state in $(i+1, j')$ is available through the following relation:

$$P\{S_{i+1}^{j'}\} = V_{S_{i+1}}(j') \quad (5.7)$$

In order to initialize the computation process, an initialization vector which describes the role and the type of traffic that the target node has to send is necessary. Let V_{S_0} be the vector which represents the state of the target node at slot 0.

$$V_{S_0} = \left\{ P\{S_0^A\}, P\{S_0^B\}, P\{S_0^C\}, P\{S_0^D\} \right\} \quad (5.8)$$

The possible values of V_{S_0} are represented in Table 5.3.

Target node parameters	Value of V_{S_0}
Owner with RT traffic	$\{1, 0, 0, 0\}$
Non owner with RT traffic	$\{0, 1, 0, 0\}$
Owner with BE traffic	$\{0, 0, 1, 0\}$
Non owner with BE traffic	$\{0, 0, 0, 1\}$

TABLE 5.3: Possible values of the initialization vector V_{S_0}

5.3.2 Calculation of the probability to find the channel busy

The status of the channel when a node senses the channel determines if it may start to transmit or not. If the channel is found busy, this means that another node is already transmitting. Therefore, the node must delay its transmission, otherwise to two simultaneous transmissions would cause a collision. In AMPH, once a node gains access to the channel, it transmits as much packets as it can before the end of the slot. As a consequence, when one node starts a transmission, the other nodes will find the channel busy for the rest of the slot.

In this part, we aim to compute the probability that the target node finds the channel busy in (i, j) denoted as ${p_b}_i^j$. In order to simplify the formulation of p_b, we express this probability as the opposite of the probability that the channel is free, denoted as p_f, and $p_b = 1 - p_f$.

We compute ${p_f}_i^j$ from the point of view of the target node.The probability that the target node finds the channel free depends on the type of traffic that the target node wants to transmit, and on its role during the current slot (owner or non owner).

${p_f}_i^j$ also depends on the traffic of other sender nodes which are competing to transmit during the current slot, denoted as contenders. The probability that a contender wants to send RT traffic is denoted as p_{rt} and the probability that it wants to send BE traffic is denoted as p_{be}. The probabilities p_{rt} and p_{be} are considered to be constant over time.

We split the calculation of ${p_f}_i^j$ into four steps according to the role and the traffic type of the target node, i.e., if j belongs the contention window A, B, C or D.

- $j \in A = [0, \beta_A[$

  $j \in A$ when the target node is the owner of the current slot and has RT traffic to send. In this case, it has the highest priority to access the channel and it is impossible that another node started transmitting before, therfore ${p_f}_i^j = 1$.

- $j \in B = [\beta_A, \beta_B[$

  $j \in B$ when the target node has RT traffic to send but is not the owner of the slot. If the owner of this slot did not have RT traffic in its sending queue or had no traffic at all, the target node can still contend for channel access. However, other nodes can also have RT traffic to send and compete to access the channel.

For $j = \beta_A$, only the owner of the slot can be transmitting, so ${pf_i}^{\beta_A}$ is equal to the probability that the owner had no RT traffic to send:

$$ {pf_i}^{\beta_A} = 1 - p_{rt} $$

For the other values of j in B, the channel can be found free if the channel was already free in $j = \beta_A$ and no node started to transmit between $\beta_A + 1$ and j. The probability that at least one node started a transmission between $\beta_A + 1$ and j is equal to the probability that its backoff value was in $[\beta_A + 1, j]$ and that there was RT packets in its queue. Given that the total number of nodes in the network is equal to N, in the worst case scenario, the number of contenders in this scenario is $N - 2$ (total number of nodes minus the target node and the owner), and as $P\{b_v \in [\beta_A + 1, j]\} = \frac{j-\beta_A}{\beta_B-\beta_A}$, we have:

$$ \begin{aligned} {pf_i}^{j} &= {pf_i}^{\beta_A} \cdot (1 - p_{rt} \cdot \frac{j - \beta_A}{\beta_B - \beta_A})^{(N-2)} \\ &= (1 - p_{rt}) \cdot (1 - p_{rt} \cdot \frac{j - \beta_A}{\beta_B - \beta_A})^{(N-2)} \end{aligned} $$

- $j \in C = [\beta_B, \beta_C[$

  A node whose backoff value belongs to the interval C is the owner of the slot and does not have RT traffic. No other node can compete to have access to the channel in C, but a transmission may already be in progress if at least one of the remaining nodes had RT traffic to send. The probability that the channel is found free for $j \in C$ is equal to the probability that no other node had RT traffic to send in slot i:

$$ {pf_i}^{j} = (1 - p_{rt})^{(N-1)} $$

- $j \in D = [\beta_C, \beta_D[$

  $j \in D$ means that the target node only has RT traffic to send and is not owner of the slot, therefore it has the lowest priority to access the channel. Nevertheless, it is still possible to find the channel free. The channel can be free in $j = \beta_C$ if no node had RT traffic, and if the owner of the slot did not have BE traffic either:

$$ {pf_i}^{\beta_C} = (1 - p_{rt})^{(N-1)} \cdot (1 - p_{be}) $$

For the remaining possible values of j, i.e., for $j \in [\beta_C + 1, \beta_D - 1]$, the probability that the channel is free is equal to the probability that the channel was already free in $j = \beta_C$ and no node started to transmit between $\beta_C + 1$ and j:

$$\begin{aligned} {p_f}_i^j &= {p_f}_i^{\beta_C} \cdot (1 - p_{be} \cdot \frac{j - \beta_C}{\beta_D - \beta_C})^{(N-2)} \\ &= (1 - p_{rt})^{(N-1)} \cdot (1 - p_{be}) \cdot (1 - p_{be} \cdot \frac{j - \beta_C}{\beta_D - \beta_C})^{(N-2)} \end{aligned}$$

We computed the probability that the target node finds the channel free for all possible values of j, at a generic time slot i. Since ${p_f}_i^j$ only depends on p_{rt}, p_{be}, N, and CW size, and given that all these values are constant over time, p_f is identical for every slot ($\forall\, i$):

$${p_f}_0^j = {p_f}_1^j = (...) = {p_f}_{N-1}^j$$

Given that $p_b = 1 - p_f$, the probability to find the channel busy is:

$${p_b}_i^j = \begin{cases} 0, & \text{for } j \in A \\ 1 - (1 - p_{rt}) \cdot (1 - p_{rt} \cdot \frac{j - \beta_A}{\beta_B - \beta_A})^{(N-2)}, & \text{for } j \in B \\ 1 - (1 - p_{rt})^{(N-1)}, & \text{for } j \in C \\ 1 - (1 - p_{rt})^{(N-1)} \cdot (1 - p_{be}) \cdot (1 - p_{be} \cdot \frac{j - \beta_C}{\beta_D - \beta_C})^{(N-2)}, & \text{for } j \in D \end{cases} \tag{5.9}$$

5.3.3 Calculation of success probability

The success probability p_s is the probability that the target node successfully transmits a packet to the base station, i.e., that the node succeeds in accessing the channel and no collision occurs during the transmission. Collisions may occur if two transmissions start at the same time, which happens when two nodes get the same backoff value and sense the channel simultaneously. In order to compute p_s, we compute the probability that the transmission started by the target node in (i, j) is unique, denoted as ${p_u}_i^j$.

${p_u}_i^j$ is equal to the probability that no contender gets the same backoff value as the target node. We compute p_u according to the possible values of j.

- For $j \in A$ and $j \in C$

  In this case, since the target node is the owner of the slot, it is the only one that can compete for channel access during these intervals. As a result, we have:

$$p_{u_i}^{j} = 1$$

- For $j \in B$

  Only nodes with RT traffic to send can get a backoff value in this interval apart from the owner of the slot.

  Let be G_i the event "the i^{th} contender gets the same backoff value as the target node". The sample space is $\Omega = B$ and $|\Omega| = \beta_B - \beta_A$. We have:

$$P\{G_i\} = p_{rt} \cdot \frac{1}{\beta_B - \beta_A}$$

  The probability $P\{G_i^C\}$ that the i^{th} contender does not get the same backoff value as the target node is $1 - P\{G_i\}$. In order to get the probability that the transmission of the target node is unique, none of the contenders should pick this value. The probability that no contender picks the same backoff value as the target node is:

$$p_u = (1 - p_{rt} \cdot \frac{1}{\beta_B - \beta_A})^{(N-2)}$$

- For $j \in D$

  The transmission attempt is unique if no contender got the same backoff value as the target node, in the event that the contenders had BE traffic and no RT traffic (otherwise their backoff would have been in B). We use the same method as above, the only difference is the sample space is $\Omega = D$ and $|\Omega| = \beta_D - \beta_C$. Also, only nodes with BE traffic and no RT traffic can be contenders. We have:

$$p_u = (1 - p_{rt})^{(N-1)} \cdot (1 - p_{be} \cdot \frac{1}{\beta_D - \beta_C})^{(N-2)}$$

The probability that the transmission of the target node is unique is:

$$p_{u_i}^{\ j} = \begin{cases} 1, & \text{for } j \in A \\ (1 - p_{rt} \cdot \frac{1}{\beta_B - \beta_A})^{(N-2)}, & \text{for } j \in B \\ 1, & \text{for } j \in C \\ (1 - p_{rt})^{(N-1)} \cdot (1 - p_{be} \cdot \frac{1}{\beta_D - \beta_C})^{(N-2)}, & \text{for } j \in D \end{cases} \tag{5.10}$$

Finally, we obtain p_s using the following relation:

$$p_s = \sum_{i=0}^{N-1} \left(\sum_{j=0}^{\beta_D - 1} \left(P\{T_i^j\} \cdot p_{u_i}^{\ j} \right) \right) \tag{5.11}$$

5.3.4 The Algorithm

In order to compute all the target performance metrics, we provide the following algorithm. It allows the evaluation of the probability that the target node starts a transmission within a given time and the associated probability of success, according to the following parameters. Through these parameters, we will subsequently analyze the performance of our protocol in depth under various scenarios.

- The type of traffic of the target node
- The traffic of contenders p_{rt} and p_{be}
- The network size N

Id and V_{S_0} are initialization data. Id is the identifier of the target node, and also the slot number to which it is assigned. V_{S_0} describes the initial state of the target node and is initialized according to Id and the type of traffic that we aim to study. We provided the possible values of V_{S_0} earlier in Table 5.3.

Algorithm 1

Input: N, Id, p_{rt}, p_{be}, and V_{S_0}
Output: $P\{T\}$ and p_s
  Compute ${p_b}^j \; \forall \; j$ according to (5.9)
  Compute $Trans_1$, $Trans_2$, and $Trans_3$ according to (5.4), (5.5), and (5.6)
  Compute $P\{T_0^j\} \; \forall \; j$ according to (5.1)
  for $i = 0 \rightarrow N - 1$ **do**
    for $j = 0 \rightarrow \beta_D - 1$ **do**
      Compute $P\{S_i^j\}$ according to (5.7)
      Compute $P\{T_i^j\}$ according to (5.1)
      Compute ${p_u}_i^j$ according to (5.10)
      Compute $V_{S_{i+1}}$ according to (5.2) and (5.3)
    end for
  end for
  Compute p_s according to (5.11)
  return $P\{T\}$ and p_s

Our algorithm assumes that the target node starts the transmission process at slot 0. However, in real operation the transmission process starts as soon as the packet is received from upper layers so the slot number can be any value between 0 and $N - 1$. Since the probability of transmission and the probability of success depend if the target node is owner or not, the performance result are highly influenced by the slot number assigned to the target node.

For instance, if we consider that the target node is the owner of slot 0 and that it wants to transmit RT traffic, $P\{T_0\} = 1$. However, if the target node is non owner, $P\{T_0\}$ falls, since it has to contend with the other nodes to gain access to the channel.

In order to evaluate the average performance, we run the algorithm with each possible value of Id and compute the arithmetic mean of $P\{T\}$ and p_s.

5.4 Numerical results and performance analysis

In this section, we use the model to analyze the performance of AMPH through the study of the behavior of one node in a given network of N nodes. The numerical application is carried out using Matlab. Equivalent scenarios are performed in the simulator OMNeT++ in order to validate the model. As explained in the previous section, the following results are the average obtained by running Algorithm 1 with each possible value of Id.

Since we aim to demonstrate the efficiency of the QoS mechanisms of our protocol, in a first phase, we study the performance of AMPH for the RT traffic class. First, we analyze the transmission probability and we derive the MAC latency for RT traffic, then we assess the data delivery ratio of RT traffic through the probability of success. The performance of AMPH regarding BE traffic is discussed subsequently.

5.4.1 Transmission probability and latency of RT traffic

In Figs. 5.7 and 5.8, we plotted the probability of transmission $P\{T\}$ as a function of i, for different values of p_{rt} while N was set to 8. Fig. 5.9 represents the corresponding cumulative function $F\{T_i\}$. As for Fig. 5.10, it plots the probability of transmission at the first attempt as a function of p_{rt} for different values of N.

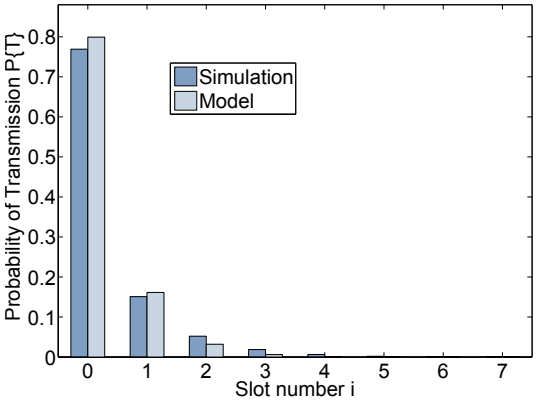


FIGURE 5.7: Probability of transmission at the i^{th} attempt (slot) where $p_{rt} = 0,07$

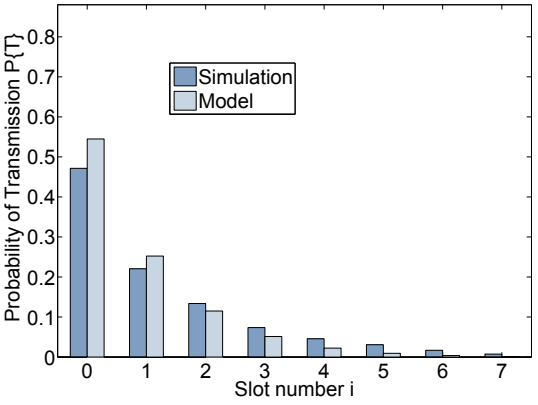


FIGURE 5.8: Probability of transmission at the i^{th} attempt (slot) where $p_{rt} = 0,19$

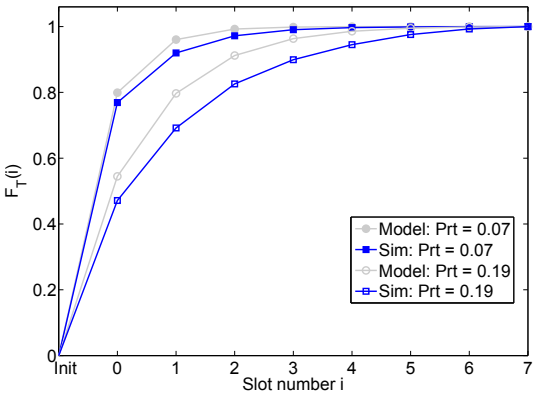


FIGURE 5.9: Cumulative function $F_T(i)$ as a function of i obtained through simulations and through the mathematical model for different values of p_{rt}.

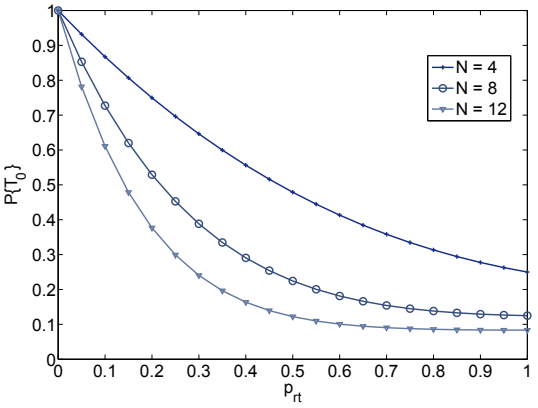


FIGURE 5.10: Probability of transmission at the first attempt $P\{T_0\}$ $(i = 0)$ as a function of p_{rt} obtained through the mathematical model for different values of N.

First, we can observe that simulation results and the mathematical model do not present significant differences, therefore the model is validated. Secondly, the cumulative function shows that the probability of transmission reaches one for $i = N - 1$, thus demonstrating that AMPH ensures that when a node has RT traffic to send, it will succeed in accessing the channel before a time frame has elapsed, in the worst case after $N-1$ attempts. Also, the cumulative function indicates that AMPH minimizes the channel access delay. Indeed, the probability of transmission is high as from low values of i. Finally, Fig. 5.10 shows how AMPH ensures high probability to transmit at the first attempt, even if $P\{T_0\}$ decreases as p_{rt} increases. Nevertheless, the drop is not sharp for low values of N. Indeed, for $N = 8$ and $p_{rt} = 0,19$, $P\{T_0\} \approx 0,5$, which is very satisfactory. In addition, this figure points out that $P\{T_0\}$ is bounded, as for $p_{rt} = 1$, $P\{T_0\} = 1/N$.

In summary, the results show that AMPH guarantees that the latency of RT traffic is minimized and bounded by the duration of one time frame. This analysis confirms the trend observed earlier by the simulation results.

5.4.2 Success probability of RT traffic transmissions

The following figures depict the results of the evaluation of the probability of success p_s obtained through simulations and the mathematical model for RT traffic.

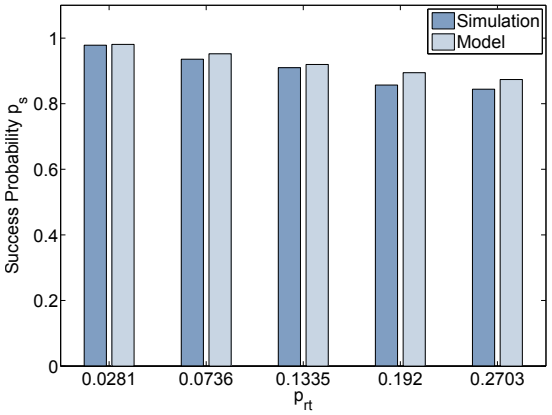


FIGURE 5.11: Transmission success probability of a RT packet obtained through simulations and the mathematical model for different values of p_{rt} and $N = 8$.

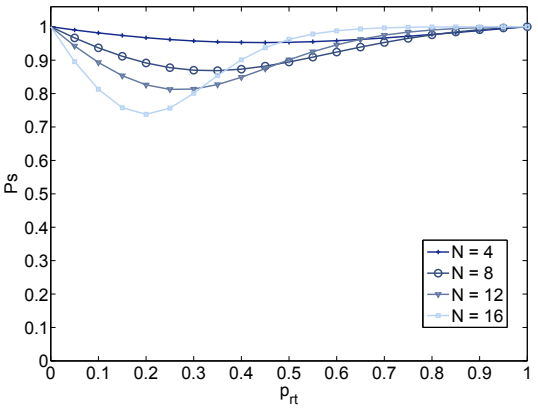


FIGURE 5.12: Transmission success probability of a RT packet as a function of p_{rt} obtained through the mathematical model for different values of N.

In Fig. 5.11, we plotted simulation results of p_s and corresponding numerical results obtained through the mathematical model in order to evaluate the data delivery ratio of AMPH. In this experiment, we fixed N to 8. As would be expected, the success probability decreases as the probability that contenders have RT traffic increases. Indeed, the probability that a neighboring node tries to send RT traffic during an empty slot increases, therefore the probability of collision rises accordingly. We observe a very light shift between the analytical and experimental values which can be explained by the random nature of our protocol and of packet arrival times, thus our model is correct.

In Fig. 5.12, we plot different values of p_s obtained through the mathematical model as a function of p_{rt} for different values of N, in order to study the influence of the traffic of contenders and of the network size on the data delivery ratio. In accordance to what we observed in the previous figure, p_s decreases as p_{rt} rises, but surprisingly, it increases past some value of p_{rt}, which varies depending on the size of the network. In fact, this behavior is normal given that when p_{rt} grows, it is more likely that other nodes have RT traffic to send during their own slot, thus the target node will not find empty slots to steal and will have to wait for its own time slot to perform its sending attempt, in which it is impossible that a collision occurs.

We notice that the reliability of AMPH deteriorates as N grows for medium values of p_{rt}. Indeed, the larger the network is, the more the reliability decreases, as p_u decreases exponentially as a function of N. Nevertheless, if we look at these results from the temporal point of view, the collision probability can also be seen as the probability that the target node accesses the channel without waiting for its own slot, thus improving latency. There is a trade-off between reliability and latency. In our solution, we chose to focus on latency, since high-throughput traffic like multimedia traffic is relatively loss tolerant but not delay tolerant. Also, the reliability in small networks or for low values of p_{rt} remains fully acceptable: below a 10% packet loss, coding techniques can compensate [38]. If one may want to use AMPH under unfavorable conditions, in order to overcome this problem, it is entirely possible to implement a safe mode that would be triggered when excessive degradation of the reliability occurs, where the base station sends acknowledgments when the sender transmits during the time slot of another node. We decided not to implement this technique as in addition to minimizing the latency as we also aim to maximize the throughput and not to waste it by using multiple control messages.

5.4.3 Transmission probability and latency of BE traffic

In order to provide a comprehensive overview of the performance of AMPH, we also evaluate $P\{T\}$ and p_s for BE traffic. As a first step, we consider a network with no RT traffic.

In Fig. 5.13, we study the probability of transmission of BE packets by representing the cumulative function $F_T(i)$ of simulation and analytical results of the evaluation of $P\{T_i\}$ for different values of p_{be}. During this experiment, N was equal to 8. We can see that the results obtained through simulations and the mathematical model are very close, so this means that our model of the BE traffic is also correct. With no RT traffic in the network, the latency of BE traffic is similar to that of RT traffic. Indeed, the mechanism to access the channel is the same, but the overhead is bigger since the backoff values are a little bit larger.

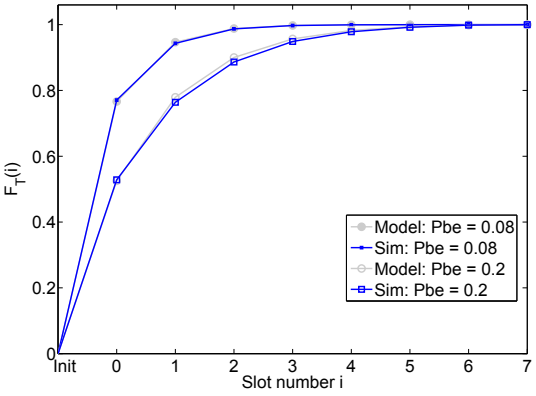


FIGURE 5.13: Cumulative function $F_T(i)$ as a function of i obtained through simulations and through the mathematical model for different values of p_{be} where $p_{rt} = 0$.

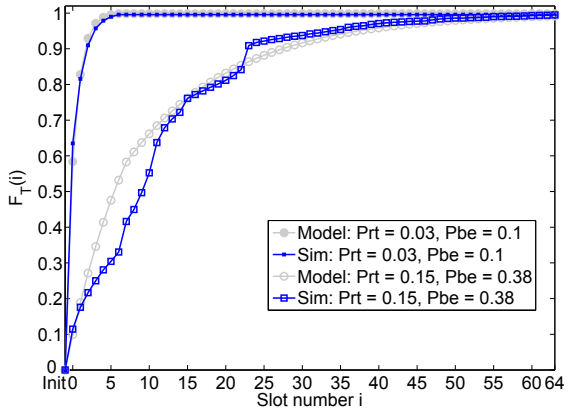


FIGURE 5.14: Cumulative function $F_T(i)$ as a function of i obtained through simulations and through the mathematical model for different values of p_{rt} and p_{be}.

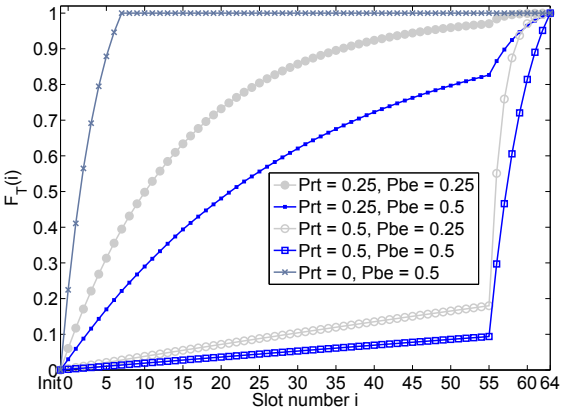


FIGURE 5.15: Cumulative function $F_T(i)$ as a function of i obtained through the mathematical model with larger values of p_{rt}.

In the following experiments, we introduce RT traffic in order to analyze its impact on the latency of BE packets. As in the previous figure, Fig. 5.14 shows the cumulative function of the transmission probability as a function of i. We still observe a good agreement between both simulation and analytical results. As p_{rt} increases, the probability to transmit BE traffic within a minimum number of time slots decreases, so the latency of BE traffic increases accordingly. However, it remains acceptable. For instance, when $p_{rt} = 0,15$, $P\{T_{25}\} > 0,8$, i.e., there is more than 80% chances that the transmission happens before $i = 25$, which gives a MAC latency of $25 * 40,96$ (slot duration) $\approx$ 1s.

In Fig. 5.15, we plot $F_T(i)$ for larger values of p_{rt}, namely $0,25$ and $0,5$, and different values of p_{be}, in order to further analyze the impact of the traffic of contenders on the MAC latency for BE packets. The results show that p_{rt} is the key parameter regarding BE latency. We can see that the parameter p_{be} also affects the results but its influence on $P\{T\}$ is smaller. For high values of p_{rt}, the transmission probability of BE packets is poor. In this case, the anti-starvation mechanism implemented in AMPH is highly desirable in order to increase the transmission probability of BE packets.

Fig. 5.15 also demonstrates the importance of our anti-starvation mechanism regarding the latency of best-effort traffic. When p_rt is high, the chances to transmit BE traffic drop. Our mechanism allows that in M frame among N, BE traffic has priority over RT traffic. In this experiment, N was set to 8 and M to 1, so one frame among eight is arranged to favor BE traffic over RT traffic. Since in a star topology, the maximum number of slots is equal to the number of nodes in the network, the size of the frame is equal to 8. The switch in the priorities occurs at the 8^{th} frame, then 56 slots have elapsed (7 x 8 slots). We can see the discontinuity in the figure at $i = 56$, from where the transmission of BE traffic is favored. At the end of this frame, we notice that $F_T(i)$ reaches 1, thus proving that the latency of BE packets is also bounded. As a consequence of our anti-starvation mechanism, the maximum MAC latency for BE traffic is $N^2 \cdot slot\ duration$. This mechanism is optional and may be triggered only when the BE queue reaches a certain threshold, and the occurrence frequency of special frames can be adjusted according to the traffic conditions through the parameter M. However, the more often the special frame occurs, the less bandwidth remains for RT traffic. A trade-off must be found between BE latency, RT latency, and the throughput required by each traffic class.

5.4.4 Success probability of BE traffic transmissions

Finally, we consider the probability of success when transmitting BE packets. In Fig.5.16, we plotted p_s regarding BE traffic for different values of p_{be}, $N = 8$ and $p_{rt} = 0$. The results were obtained through simulations and using the model. We observe the same behavior as for the probability of success of RT traffic transmissions. As p_{be} increases, p_s decreases, until the probability to find an empty slot falls, therefore p_s rises. We can see that the success probability of BE traffic transmissions is high, which confirms that AMPH achieves high reliability for BE traffic as well, as we observed through the simulation results in Chapter 4. In the worst case, when $p_{be} = 0.2888$, p_s stays above 0.8. Considering that BE traffic is mostly redundant, the impact of a limited packet loss is negligible.

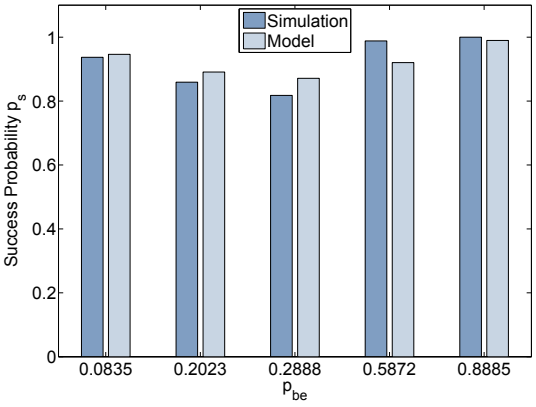


FIGURE 5.16: Transmission success probability of a BE packet obtained through simulations and the mathematical model for different values of p_{be} and $N = 8$.

5.4.5 Discussion / Conclusion

In this chapter, we proposed a mathematical model of our MAC protocol in order to deeper analyze its performance. We demonstrated that through our slot ownership and stealing mechanism, the latency of RT packets is minimized and bounded by the duration of one time frame, and that our anti-starvation mechanism allows that the latency of BE packets is also bounded In addition, we showed that the success probability of a transmission is high for a moderate number of contenders ($Ps \geq 0.8$ when $N \leq 12$), so AMPH achieves high reliability. These observations confirm the good results obtained through simulations.

As a future work, we aim to use the model to evaluate additional performance parameters as channel utilization and protocol overhead. However, in order to obtain accurate results when evaluating the transmission of several packets, we need to better model the traffic of neighboring nodes. For now, the probability that neighboring nodes have real-time traffic is fixed, and we assume that all neighbors contend to access the channel (worst case scenario), which is not exact. We intend to enhance the model by using queuing theory, and to model incoming traffic with probability distribution functions.

Chapter 6

Adding Multihop Support to AMPH

In order to provide QoS at the MAC layer to heterogeneous wireless sensor networks, we proposed a new MAC protocol called AMPH. AMPH performs very well when used in star topology networks, however, it suffers from the hidden terminal problem in multihop scenarios. We aim to improve our protocol so that it can also be used in multihop topologies, which are employed by many WSN applications. In this chapter, we present our solution to provide multihop support to AMPH. We explain in detail the new contention mechanism and we provide some performance evaluation results.

6.1 A new slot-stealing mechanism

Using AMPH in multihop scenarios raises problems because of the contention in the slot-stealing process. Indeed, nodes that contend to steal the slot of a common neighbor may not hear each other and start simultaneous transmissions leading to a collision. Therefore, the contention should be organized. We propose a new contention mechanism where the owner of the unused slot coordinates the contention between the neighboring nodes. The owner sends a message that indicates which neighbor is authorized to transmit. When receiving this message, the designated node is informed that it is allowed to steal the slot, while the other neighbors will deduce that they have to remain silent for the end of the current time slot.

In Fig. 6.1, we show a multihop scenario where AMPH leads to collisions. We assume that the current slot is slot 4. Node W, which is the owner of slot 4, has no data to send, therefore nodes A and B contend for channel access. Since A is not in range of B, it is not able to hear if B is

transmitting and vice versa. As a result, both A and B start to transmit and a collision happens at the receiver S.

Fig. 6.2 illustrates our new contention mechanism. In this new version of AMPH denoted as AMPH v2, the owner of the unused slot designates one neighbor to use the slot by sending a message. While W sends a message to A to inform it that it is allowed to steal his slot, B receives the message as well. It is thus informed that A has been designated to transmit during the current slot and that consequently it must remain silent. Through this mechanism, no colliding transmissions occur.

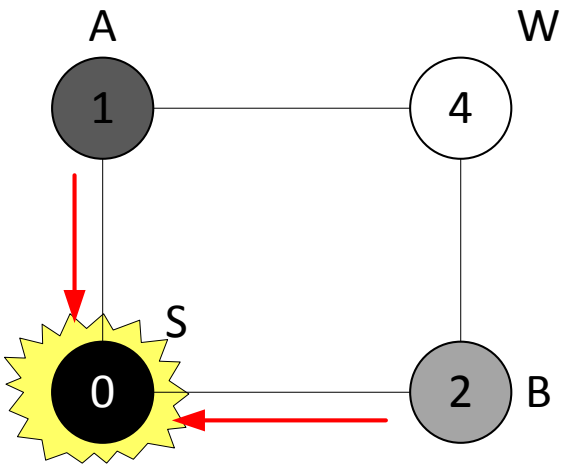


FIGURE 6.1: Hidden terminal problem in AMPH

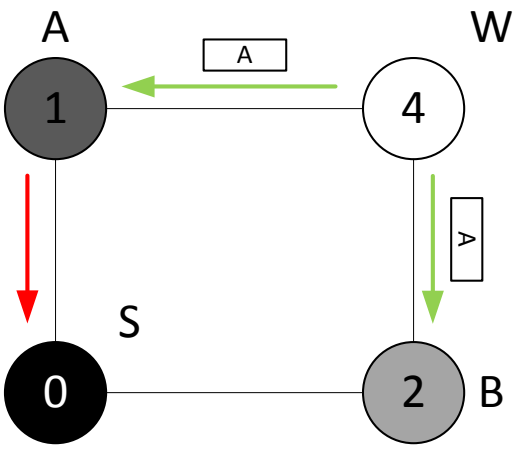


FIGURE 6.2: Solving the hidden terminal problem by sending a control message

The performance of our mechanism depends on the method used by the owner to designate the neighbor that will be allowed to steal the slot. The simplest way to organize the contention consists in designating all neighbors in turn, i.e., in inviting them one after another to send their data. However, this method may lead to an inefficient use of the channel. The difficulty of the scheme of our new solution lies in the fact that the coordinator does not know which nodes have data to send and may want to steal his slot. When the owner allocate its unused slot to nodes that do not have data to send, the slots remain unused and the bandwidth is wasted. In order to reduce the probability of polling nodes that do not have data to send, we propose an adaptive method where the coordinator elects the neighbors according to their transmission history during the previously stolen slots. Nodes that effectively transmitted data when they were polled will be given higher chance to be elected in the next frames than nodes that did not have data to send. Through this adaptive mechanism, nodes with high traffic loads have more transmission

opportunities than nodes with less traffic. As a result, the latency of packets from loaded nodes is improved and the channel utilization is increased.

6.2 Definition of the election process

The principle of our new mechanism is that owners that do not have data to send during their reserved slot invite the potential neighboring senders to transmit their data, so this slot is not left unused. In order to ensure that the slot will be effectively used, the owners have to choose neighbors that have data to send. Since nodes are not able to know if their neighbors have data to send, our method is probabilistic. The probability that the owner of an unused slot invites a given neighbor to steal its slot is based on its past uses of unused slots. In what follows, we explain in details the election process.

We consider that all nodes maintain a list of their one-hop neighbors, and each neighbor is assigned a probability p. p is the probability that a given neighbor is designated to transmit during the empty slot. At the beginning of the process (at startup), all probabilities are equal. If the number of neighbors is N, all probabilities p_i are equal to $\frac{1}{N}$. In order to designate which neighbor is allowed to use their slot, owners that do not have data to send perform a random selection according to the weight of each neighbor. We call this process the lottery and the designated node is denoted as the winner of the lottery.

During the protocol operation, all nodes record how many times their neighbors win the lottery and also how often they effectively use the slots that they won. We update the probabilities once all neighbors of a given owner won the lottery at least M times (i.e., they had at least M opportunities to transmit during the unused slot of the owner). For each neighbor i of a given node, we compute the ratio of used slots denoted as r_i:

$$r_i = \frac{\text{Number of slots used}}{\text{Number of slots won}}$$

Afterwards, the new probability to win the lottery p_i is set proportionally to the ratio of used slots of all neighbors:

$$p_i = \frac{r_i}{\sum_{i=1}^{N} r_i}$$

In order to guarantee a fair lottery draw, no node should reach a probability equal to zero, otherwise the node in question would not be able to be participate to the lottery anymore. To avoid this situation, if one node did not use any slot won, we still set its number of slots used to one.

6.3 Performance evaluation

We aim to demonstrate that our adaptive protocol adapts to the traffic load through simulation experiments in OMNeT++. We consider a network of five nodes A, B, C, D, and Z, and a base station S, whose topology is provided in Fig 6.3. The dashed lines indicate the connections between all nodes in the network. The solid lines are active communication links for data transmissions. The picture also shows the slot number assigned to each node.

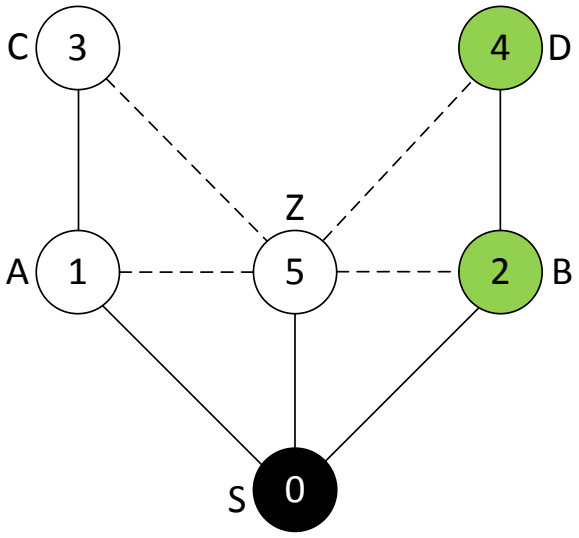


FIGURE 6.3: Multihop topology for the performance evaluation of AMPH v2

We set an heterogeneous traffic configuration. All nodes have real-time traffic, but node D has a higher traffic load. The frame rate of nodes A, B, C and Z is set to 2 fps and the frame rate of D is 10 fps. It should be noted that the traffic of upstream nodes adds up to that of relay nodes. For instance, the traffic load of B is 12 fps since it has to relay the traffic from D. We study how the slots 3 and 5, owned respectively by node C and node Z, are used when they have no data to send.

Node Z is a central entity in this network. All nodes participate in the lottery to win its slot, excepted the base station S since we assume that it does not send any data. In Fig 6.4, we show the progression of the winning probability of each neighbor of node Z. At startup, all probabilities are equal. After the first update, we can see that the probabilities immediately adapt to the traffic load of each neighboring node. As nodes B and D has a significant higher traffic load than nodes A and C, our adaptive algorithm gives them higher chance to win the slot of node Z.

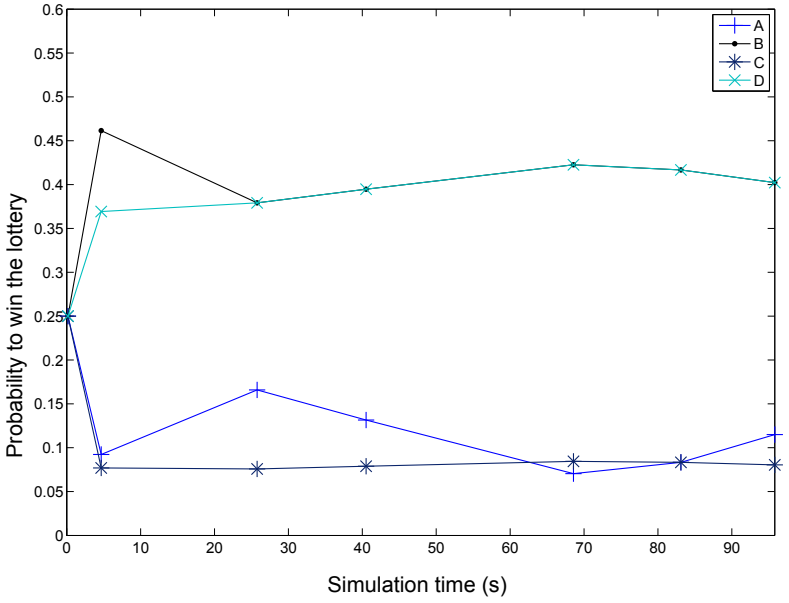


FIGURE 6.4: Progression of the probability to win slot 5

In Fig 6.5, we show the progression of probability to win slot 3 owned by node C. Node C only has two neighbors, node A and node Z, which have similar traffic loads. Accordingly, they should have equal winning probabilities. We can see that the probabilities are effectively centered on 0.5 but that they oscillate. This behavior is due to the varying traffic loads of A and Z. For instance, Z may not need to use slot 3 if it already transmitted during slot 5 or slot 1. Also, since our method is probabilistic, oscillations may come from a non-uniform selection during a given update window. In order to reduce the oscillations, we may extend the update window by increasing M. However, the algorithm would be less reactive to traffic changes.

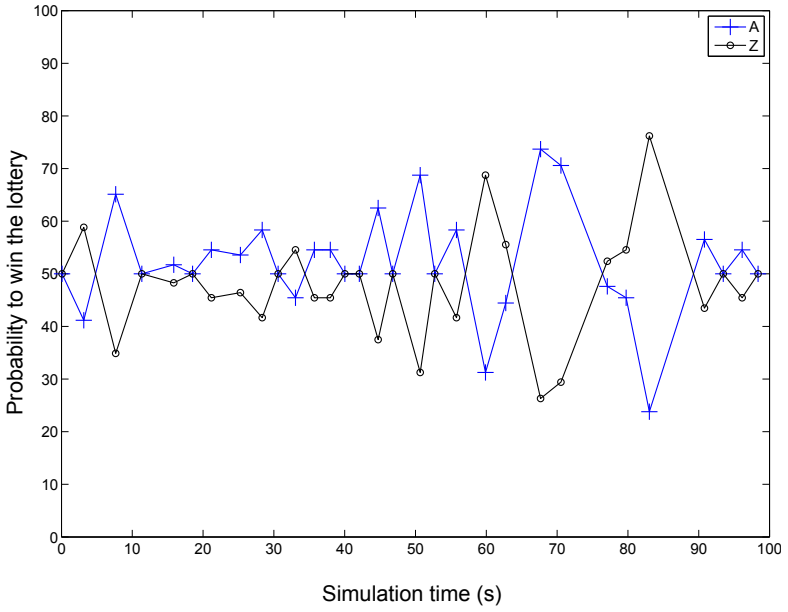


FIGURE 6.5: Progression of the probability to win slot 3

6.4 Conclusion

In this chapter, we presented a new version of AMPH fitted to multihop topologies. We introduced a new contention mechanism which solves the hidden terminal problem and adapts to the traffic load. Through simulation experiments, we were able to verify that our protocol effectively gives higher chance to access the channel to nodes with high traffic loads. In addition, under uniform traffic loads, our protocol gives a fair access to each node. Through our new contention mechanism, AMPH v2 still efficiently uses the bandwidth and allows to reduce the latency. As a future work, we intend to further analyze to performance of AMPH v2 in terms of channel utilization, latency and reliability, and to compare it to that of Diff-MAC.

Chapter 7

Experiments on the Imote2 platform

After having validated the performance of our MAC protocol for heterogeneous WSN through simulation experiments and a mathematical model, we aimed to implement and operate sensor network applications in order to show the benefits of AMPH in real situations. In this chapter, we describe the hardware and software aspects of the selected platform, then we go through the implementation of AMPH and of a realistic test application. Finally, we discuss the performance of AMPH through experimental results.

7.1 Goals

We aim to demonstrate the feasibility and the performance of our research work through the implementation of AMPH on real sensor nodes and the execution of realistic application scenarios. Multimedia applications are typical heterogeneous WSN applications, with various sensing capabilities and traffic types, and high QoS requirements. We opted for the Imote2 platform to set up this application since it is a modular stackable platform which can be customized with extension boards. In particular, multimedia capabilities can be added to Imote2 nodes through an extension board equipped with a camera. In addition, the Imote2 platform is supported by TinyOS, a widespread operating system for low-power wireless embedded systems. Since the laboratory already worked with platforms running under TinyOS, it was also important to find platform supported by this operating system for compatibility and knowledge capitalization reasons.

7.2 Platform description

7.2.1 Hardware components

An Imote2 node is composed of modular and stackable boards: a processor board, which is the core of the node, an extension board equipped with sensors, and a battery board that provides energy to node. Different extension boards provide various sensing capabilities allowing the platform to adapt to a wide range of applications. A brief description of each component of the platform is provided below, along with pictures of some boards in Figs. 7.1 and 7.2.

The Imote2 is built around the low power PXA271 XScale CPU and contains an 802.15.4 compliant radio (TI CC2420) with an integrated antenna. This powerful processor includes a wireless MMX coprocessor to accelerate multimedia operations, but it can also operate in a low voltage (0.85V) and a low frequency (13 MHz) mode, hence enabling low power operation. Three controllable LED are present on the processor board. However, there are no sensors on this board. In order to add sensing capabilities to a node, the processor board has to be coupled with an extension board. The ITS400 sensor board contains several sensors thus providing multiple sensing capabilities for the Imote2 and allowing a wide variety of applications. It includes a three-axis accelerometer, a temperature/humidity sensor, a light sensor and a 4 channel A/D converter. The IMB400 is a rich board which adds multimedia capabilities to the Imote2. It is equipped with a camera and a microphone/speaker which allow the capture of still images or movies and sound playing/recording. This board also includes a motion sensor which enables movement detection. This feature allows nodes to wake-up from sleep if movement is detected. The battery board includes three slots for 1.5 V AAA batteries. A switch allows to turn the system on and off. Finally, the interface board allows programming and debugging for the Imote2. It provides both an USB serial port and a JTAG interface.

We purchased a set of nodes including basic sensor boards and a multimedia board in order to design a small wireless multimedia sensor network and run multimedia applications. Due to its various sensing capabilities, the Imote2 platform allows the development and evaluation of a wide range of wireless sensor network applications. The powerful processor board allows data-rich computations such as video processing as well as low-power operation for applications which require a long lifetime. The multifunction multimedia board enables the implementation of advanced multimedia applications such as intrusion detection and target tracking.

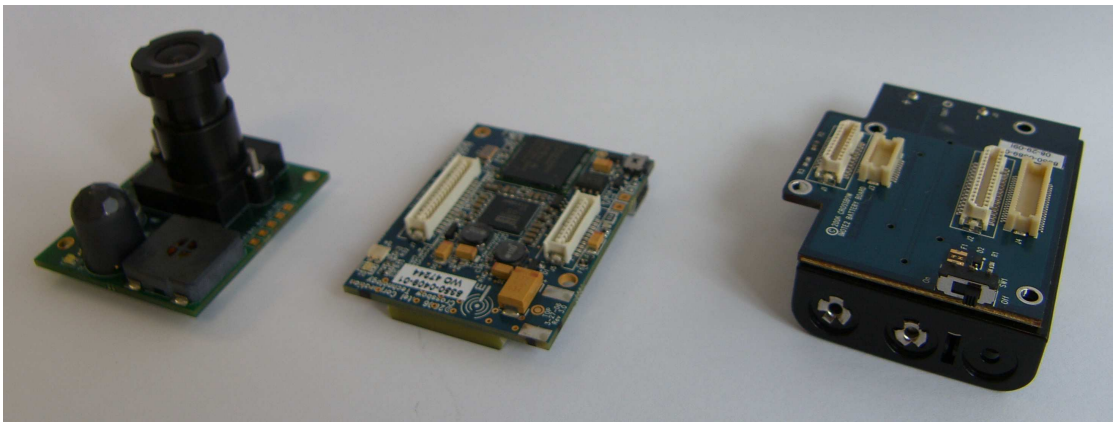


FIGURE 7.1: Multimedia board - Imote2 processor board - Battery board

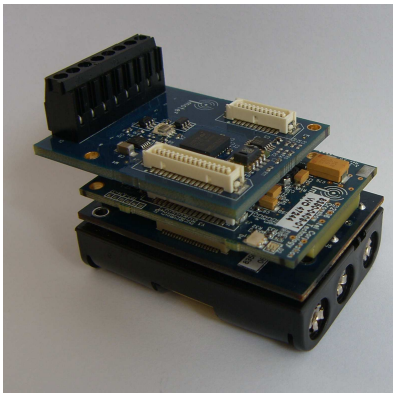


FIGURE 7.2: Stack of boards forming a basic wireless sensor node

7.2.2 Software

The Imote2 platform runs under TinyOS. It is a free and open source operating system designed for low-power wireless embedded systems such as wireless sensor networks. TinyOS system and applications are written in nesC, which stands for network embedded systems C, a dialect of the C language. Programs are built out of components, which are assembled to form whole programs. Components provide a set of interfaces and they may be linked to other components so they can use their interfaces. TinyOS provides interfaces and components for common abstractions such as packet communication, routing, sensing, actuation and storage. In Fig. 7.3, we show how an application module uses interfaces from other components. Blink is a simple application which displays a counter on the three LEDs of the processor board, and BlinkC is the main component of the application Blink. It is linked to three components: MainC, TimerMiliC and LedsC. MainC provides the interface Boot, which signals that the system has successfully booted. TimerMiliC is a timer abstraction and provides millisecond granularity timers. The

application may instantiate as many timers as required. LedsC provide simple functions to control the LEDs of the node such as switch on, switch off and toggle. BlinkC is wired to these components so it can benefit from the interfaces and functions that they provide.

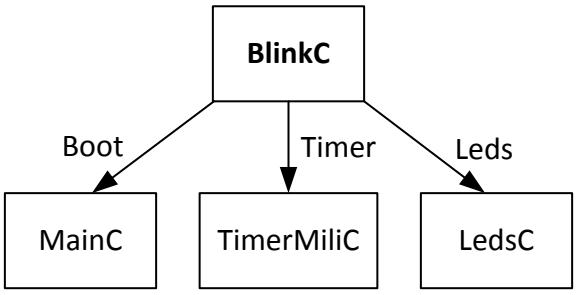


FIGURE 7.3: Components used by the BlinkC module

7.2.3 Getting started with the Imote2 platform

Setting up of the programming environment for the Imote2 platform consists of three steps: TinyOS tree installation, cross compiler installation for the Imote2 processor, and a programmer able to flash (program) motes and communicate with them. The cross compiler allows to create executable code targeted for the Imote2 platform although the program is developed and built under another environment. Programming and debugging nodes via the interface board necessitate a tool named Open On-Chip Debugger (OpenOCD) which provides drivers for the JTAG interface. Since few documentation on the Imote2 is provided by the manufacturer, it was not easy to install the developing environment and to be able to flash nodes and run simple programs. After having gathered information on the different installation steps through several sources on the Internet, we managed to successfully install all the required components. In order to facilitate the task for future users, we wrote a detailed tutorial on the installation of the tool chain required for the development and the compilation of applications for the Imote2 platform. We also explain how to compile Blink and program a node with the resulting executable, and how to set up and run a demo application which captures still images using a wireless multimedia node, transmit them to a base station, and displays the pictures on the screen of the computer. Our tutorial is available online on the Heudiasyc laboratory website [39].

7.3 Implementation of AMPH

The implementation of AMPH only requires the use of basic components. Indeed, its operation is simple and mainly consists in transmitting or receiving packets, and waiting. The general design of the AMPH module named AMPHDemoC is illustrated in Fig. 7.4. It shows the links between our module and the required components. AMPHDemoC uses timers from the component TimerMiliC, and the component ActiveMessageC which provides appropriate interfaces for radio communications: Packet, Send, and Receive. QueueC is a general FIFO queue component that allows to put and remove packets through practical functions. Finally, AMPH used LedsC and MainC to control the LEDs and the boot sequence.

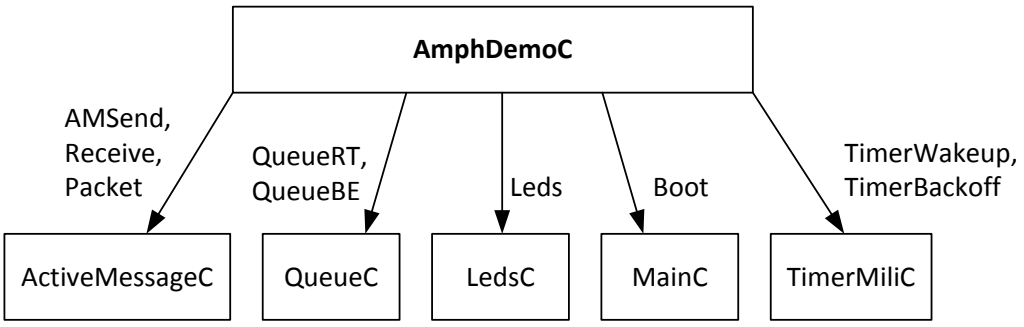


FIGURE 7.4: Components used by AmphDemoC

Our protocol requires the use of two timers. TimerWakeup defines the time structure employed in AMPH operation: this periodic timer has a period of one slot and marks the beginning of each time slot. TimerBackoff is a one-shot timer that implements the backoff operation: it is started at the beginning of the time slot when the node has data to send, for a random duration according to the role of the node (owner or non owner) and the type of traffic to send (RT or BE). When the timer fires, the node may transmit if there is no ongoing transmission. Since AMPH supports two types of traffic: real time (RT) and best effort (BE), AMPHDemoC instantiates two sending queues: one for each traffic type. Incoming packets are placed in the appropriate queue according to their priority which is set in a reserved field of the packet header. We implemented a strict priority scheduler as a function which selects the next packet to send. The function returns RT packets as long as the queue is not empty, then it returns BE packets. This function is called at the beginning of each time slot to check is there are packets to send, and when a packet has been sent in order to set the next packet to send if the current slot is not over.

Since the implementation of our protocol is only intended for demos, we did not implement any synchronization protocol. The protocol is started upon the reception of a one synchronization message send by the base station. No other message is subsequently sent. We assume that the duration of the demos is short enough such that the clock drifted is negligible.

7.4 Development of an application of intrusion detection

On top of the component AMPHDemoC, we implemented a simple intrusion detection application. This demo application aims to monitor environmental parameters such as temperature and humidity, and to detect intrusions into a given area. This application requires that intrusion alerts are transmitted in real-time. Our network is composed of four nodes: two nodes equipped with temperature sensors, one node with a motion sensor, and a base station linked to a computer that serves as a monitoring station. They are arranged in a star topology. The temperature is periodically measured and the readings are immediately sent to the base station. The node equipped with the motion sensor send alarms to the base station when an intrusion is detected. We consider that temperature readings are best-effort traffic and intrusion alarms are real-time traffic.

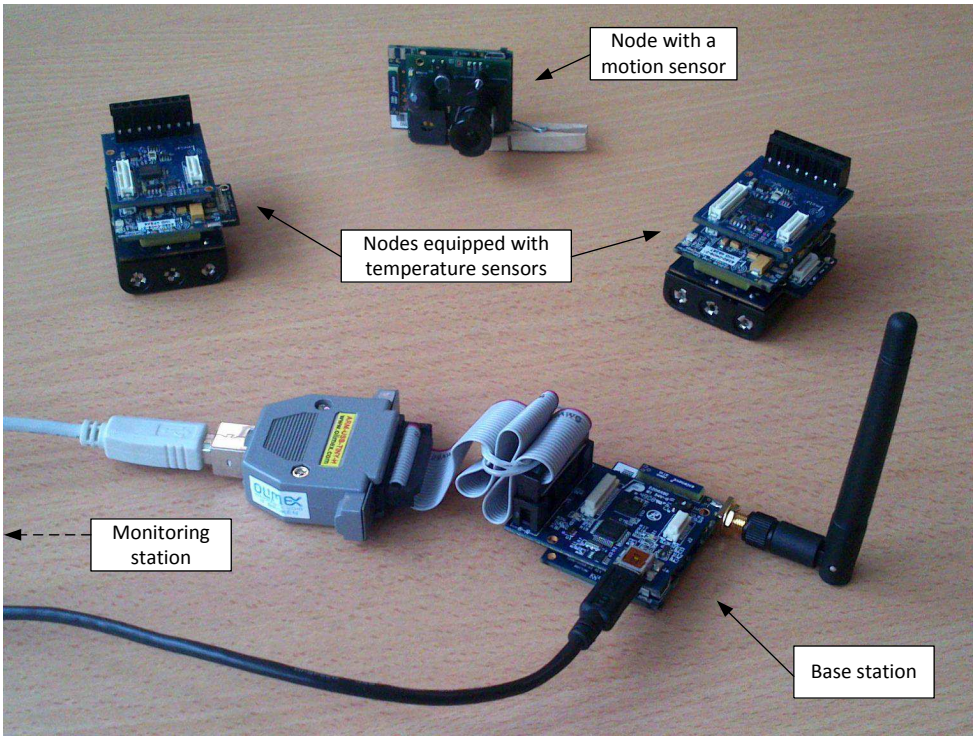


FIGURE 7.5: Our wireless sensor networks for intrusion detection

7.5 Experimental results

In this section, we present early results of the experimental evaluation of AMPH. For the moment, we have not performed extensive tests and performance evaluation. However, we were able to verify the operation of AMPH through the simple application of intrusion detection presented in the previous section. We have run the application on our platform and we recorded the packets received by the base station. In Table 7.1, we present an excerpt of the data received by the base station during a short run of our demo application. This recording shows the behavior of AMPH with and without the presence of real-time traffic.

Nodes 2 and 3 are equipped with a temperature sensor. They periodically send their measurements to the base station. The raw data is shown in the column "Value", and the corresponding temperature is computed in the column "Temperature". The sampling frequency of temperature measurements is set such that the overall data rate at the base station is medium. The mean packet arrival rate is 1.25 packets per time slot. Node 1 is only equipped with a motion sensor. When the sensor is triggered, the node sends alarm messages containing an arbitrary value (55555). A flag is set in the packet header which indicates that these packets are real-time traffic.

From slot 1 to 12, nodes 2 and 3 normally send the temperature readings. Then the motion sensor is triggered and alarm messages are transmitted from slot 13 to 16. Once the transmission of real-time alarms is over, nodes 2 and 3 transmit the queued messages during their reserved time slot. Finally, the normal operation resumes.

The recordings show that nodes transmit during any time slot, they do not have to wait until their reserved time slot to transmit. When the motion sensor is triggered, the node reports a number of alarm messages, which are effectively sent ahead of the best-effort traffic. Once the transmission of real-time traffic has stopped, the medium is again fairly shared between nodes with best-effort traffic. The reserved slot guarantee that all nodes access the channel during a time frame even if all nodes have data to send. This confirms that our protocol behaves according to its design principles.

Slot	Node ID	Owner		RT	Seq	Value	Temperature	Intrusion
1	2	1		0	398	6670	25,1	
2	3	2		0	312	6645	24,85	
2	3	2		0	312	6645	24,85	
3	2	3		0	400	6670	25,1	
4	3	1		0	313	6645	24,85	
5	2	2		0	401	6670	25,1	
6	3	3		0	314	6645	24,85	
7	3	1		0	315	6644	24,84	
8	2	2		0	402	6669	25,09	
8	2	2		0	402	6670	25,1	
9	3	3		0	317	6644	24,84	
10	2	1		0	403	6669	25,09	
11	3	2		0	318	6644	24,84	
12	2	3		0	404	6672	25,12	
12	2	3		0	404	6671	25,11	
13	1	1		1	178	55555		1
13	1	1		1	178	55555		1
13	1	1		1	178	55555		1
13	1	1		1	178	55555		1
13	1	1		1	178	55555		1
13	1	1		1	178	55555		1
14	1	2		1	179	55555		1
14	1	2		1	179	55555		1
14	1	2		1	179	55555		1
14	1	2		1	179	55555		1
15	1	3		1	180	55555		1
15	1	3		1	180	55555		1
16	1	1		1	181	55555		1
16	1	1		1	181	55555		1
16	1	1		1	181	55555		1
16	1	1		1	181	55555		1
16	1	1		1	181	55555		1
17	2	2		0	405	6671	25,11	
17	2	2		0	405	6673	25,13	
17	2	2		0	405	6672	25,12	
18	3	3		0	319	6643	24,83	
18	3	3		0	319	6642	24,82	
18	3	3		0	319	6643	24,83	
18	3	3		0	319	6640	24,8	
19	2	1		0	406	6669	25,09	
20	2	2		0	407	6671	25,11	
21	3	3		0	320	6640	24,8	
21	3	3		0	320	6640	24,8	
22	2	1		0	408	6671	25,11	
23	3	2		0	321	6642	24,82	
23	3	2		0	321	6644	24,84	
24	2	3		0	409	6673	25,13	

TABLE 7.1: Recording of packets received by the base station during a period of 24 slots

7.6 Conclusion and future work

As proof of concept, we implemented AMPH on the Imote2 platform and we were able to verify its operation through a demo application scenario. However, the performance of AMPH should be further analyzed through more realistic application scenarios and compared with that of close competitors. Besides, we have not been able to fully exploit the potential of the camera provided by the multimedia board. We managed to set up an application which takes pictures when the motion sensor is triggered and transmits the picture to the base station, but sending one single picture requires the transmission of a large number of packets. We aimed to realize a full intrusion detection application with real-time video streaming, but unfortunately, we lacked some knowledge in compression algorithms. As a consequence, this was left for future work.

Conclusion and Perspectives

It is now established that wireless sensor networks are a major innovation and offer new opportunities in the field of monitoring applications. Composed of distributed autonomous sensor nodes equipped to sense various phenomena, these networks allow low-cost infrastructure-less monitoring applications in situations where setting up fixed infrastructure networks is difficult of infeasible. The field of application of wireless sensor networks is very wide, from environmental control of office buildings to the detection of forest fires. Originally designed to collect environmental data in large areas, the main research challenge was to maximize the network lifetime. Recently, new applications for wireless sensor networks such healthcare and multimedia applications have emerged. These applications often have heterogeneous sensing capabilities and require that the network supports different types of QoS-constrained traffic at variable rates. However, the performance of existing solutions is inadequate in the presence of high QoS requirements and variable traffic loads. The research work presented in this thesis is motivated by the need of these applications for efficient QoS-aware communication protocols. Designing efficient protocols that provide an appropriate level of performance to these applications while coping with the limited resources of sensor networks is a challenging task.

In this manuscript, we started by providing basic knowledge on wireless sensor networks, envisioned applications, and sensor networks design and protocols. We highlighted the growth of demanding and heterogeneous applications which necessitate quality of service and heterogeneity support, then we studied QoS-aware protocols for wireless sensor networks. This analysis revealed the need for traffic-adaptive schemes that are able to provide an appropriate level of service to the different traffic types of multi-purpose applications. In order to compensate for this deficit, we proposed AMPH, an adaptive MAC protocol with QoS support for heterogeneous wireless sensor networks, and we exposed the general principles of our solution. In the design of our protocol, we combined the strengths of contention-based and contention-free channel access methods identified during the analysis of related work in order to propose a hybrid scheme

allowing high channel utilization. In addition, we introduced an efficient contention mechanism which favors high priority traffic so that the latency of real-time packets is minimized. We evaluated the performance of AMPH through simulation experiments via OMNeT++. We described our approach for implementing the simulations, and we presented the simulation results. These results demonstrate the effectiveness of hybrid approaches. The hybrid channel access method of AMPH achieves higher channel utilization than contention-based schemes, in particular under high traffic loads. Besides, AMPH provides low latency for real-time traffic, while avoiding starvation of best-effort traffic. The performance of our solution is consistent with our design goals. Afterwards, we provided a mathematical model of AMPH. Through this model, we performed a formal performance evaluation of our protocol. The results corroborates the good performance of AMPH. Finally, we introduced work in progress. We discussed the use of AMPH in multihop scenarios. Since AMPH does not employ control messages, our contention mechanism suffers from the hidden terminal. Therefore, we proposed AMPH v2 which implements an enhanced adaptive contention mechanism. The objective of our new mechanism is to solve the hidden terminal while minimizing the number of control messages so the bandwidth is not wasted. Our algorithm is probabilistic and necessitates only one control message. A coordinator triggers the transmission of neighbors according to their traffic load. Through simple simulations, we provided promising preliminary performance results. We also described our approach to implement our solution on a real sensor platform. We presented the Imote2 platform and we exposed the implementation of AMPH. We set up a network composed of few sensor nodes and, through a basic application scenario, we were able to verify the operation of our protocol.

Future directions

As a future work, we envisage to improve our mathematical model in order to allow a more precise performance evaluation of AMPH. We intend to add queuing theory to better model the traffic of contenders. Besides, we aim to further develop AMPH v2 and study its performance in large networks in terms of channel utilization, latency and reliability but also jitter, fairness and energy consumption. Our ultimate goal is to provide a full protocol suite for heterogeneous applications with high requirements. In this perspective, we aim to elaborate an efficient QoS-aware routing protocol with cross-layer optimizations. In a concern of resource optimization, the routing control messages may also serve as synchronization, slot assignment, and mobility support messages. Such a solution would enable a wide range of applications such as healthcare

and target tracking applications. Finally, we also consider the adaptation of our channel access approach to other types of wireless networks. We believe that some ideas introduced in this thesis would be of interest in the context of Internet of things and more generally for unattended infrastructure-less heterogeneous wireless networks.

Bibliography

[1] Ian F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38:393–422, 2002.

[2] D. Chen and P. K. Varshney. QoS Support in Wireless Sensor Networks: A Survey. In *Proc. of the 2004 International Conference on Wireless Networks (ICWN 2004)*, Las Vegas, Nevada, June 2004.

[3] Giuseppe Anastasi, Marco Conti, Mario Di Francesco, and Andrea Passarella. Energy conservation in wireless sensor networks: A survey. *Ad Hoc Networks*, 7(3):537–568, May 2009.

[4] T. Melodia, D. Pompili, and I.F. Akyldiz. Handling Mobility in Wireless Sensor and Actor Networks. *IEEE Transactions on Mobile Computing*, 9(2):160–173, 2010.

[5] R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture: an Overview. IETF, RFC 1633, 1994. URL http://tools.ietf.org/rfc/rfc1633.txt.

[6] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Services, 1998. URL http://tools.ietf.org/rfc/rfc2475.txt.

[7] IEEE 802.11e WG, Medium Access Control (MAC) Enhancements for Quality of Service, 2005.

[8] Hade Alemdar and Cem Ersoy. Wireless sensor networks for healthcare: A survey. *Computer Networks*, 54:2688–2710, 2010.

[9] Chonggang Wang, K. Sohraby, Bo Li, M. Daneshmand, and Yueming Hu. A survey of transport protocols for wireless sensor networks. *IEEE Network*, 20(3):34–40, 2006.

[10] Yogesh Sankarasubramaniam, Özgür B. Akan, and Ian F. Akyildiz. ESRT: event-to-sink reliable transport in wireless sensor networks. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc '03, pages 177–188, New York, NY, USA, 2003. ACM. ISBN 1-58113-684-6.

[11] Y.G. Iyer, S. Gandham, and S. Venkatesan. STCP: a generic transport layer protocol for wireless sensor networks. In *Computer Communications and Networks, 2005. ICCCN 2005. Proceedings. 14th International Conference on*, pages 449–454, 2005.

[12] Vehbi Cagri Gungor, Özgür B. Akan, and Ian F. Akyildiz. A real-time and reliable transport $(RT)^2$ protocol for wireless sensor and actor networks. *IEEE/ACM Transactions on Networking*, 16:359–370, April 2008.

[13] Mohammad Hossein Yaghmaee and Donald A. Adjeroh. Priority-based rate control for service differentiation and congestion control in wireless multimedia sensor networks. *Computer Networks*, 53:1798–1811, 2009.

[14] S. Ehsan and B. Hamdaoui. A Survey on Energy-Efficient Routing Techniques with QoS Assurances for Wireless Multimedia Sensor Networks. *IEEE Communications Surveys Tutorials*, 14(2):265–278, 2012.

[15] Tian He, J.A. Stankovic, Chenyang Lu, and T. Abdelzaher. SPEED: a stateless protocol for real-time communication in sensor networks. In *Distributed Computing Systems, 2003. Proceedings. 23rd International Conference on*, pages 46–55, 2003.

[16] K. Sohrabi, J. Gao, V. Ailawadhi, and G.J. Pottie. Protocols for self-organization of a wireless sensor network. *IEEE Personal Communications*, 7(5):16–27, 2000.

[17] E. Felemban, Chang-Gun Lee, and E. Ekici. MMSPEED: multipath Multi-SPEED protocol for QoS guarantee of reliability and. Timeliness in wireless sensor networks. *IEEE Transactions on Mobile Computing*, 5(6):738–754, 2006.

[18] Adel Ali Ahmed and Norsheila Fisal. A real-time routing protocol with load distribution in wireless sensor networks. *Computer Communications*, 31:3190–3203, 2008.

[19] Jalel Ben-Othman and Bashir Yahya. Energy efficient and QoS based routing protocol for wireless sensor networks. *Journal of Parallel and Distributed Computing*, 70:849–857, 2010.

[20] Min Chen, Victor C.M. Leung, Shiwen Mao, and Yong Yuan. Directional geographical routing for real-time video communications in wireless sensor networks. *Computer Communications*, 30(17):3368 – 3383, 2007.

[21] Texas Instrument. 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver, 2013.

[22] Ilker Demirkol, Cem Ersoy, and Fatih Alagöz. MAC protocols for wireless sensor networks: a survey. *IEEE Communications Magazine*, 44(4):115–121, 2006.

[23] A. Bachir, M. Dohler, T. Watteyne, and K.K. Leung. MAC Essentials for Wireless Sensor Networks. *IEEE Communications Surveys Tutorials*, 12(2):222–248, 2010.

[24] Pei Huang, Li Xiao, S. Soltani, M.W. Mutka, and Ning Xi. The Evolution of MAC Protocols in Wireless Sensor Networks: A Survey. *IEEE Communications Surveys & Tutorials*, 15(1):101–120, 2013.

[25] Zheng Teng and Ki-Il Kim. A Survey on Real-Time MAC Protocols in Wireless Sensor Networks. *Communications and Network*, 2(2):104–112, 2010.

[26] Petcharat Suriyachai, Utz Roedig, and Andrew Scott. A Survey of MAC Protocols for Mission-Critical Applications in Wireless Sensor Networks. *IEEE Communications Surveys & Tutorials*, pages 1–25, 2011.

[27] Sana Ullah, Bin Shen, S.M. Riazul Islam, Pervez Khan, Shahnaz Saleem, and Kyung Sup Kwak. A Study of MAC Protocols for WBANs. *Sensors*, 10(1):128–145, 2010.

[28] Navrati Saxena, Abhishek Roy, and Jitae Shin. Dynamic duty cycle and adaptive contention window based QoS-MAC protocol for wireless multimedia sensor networks. *Computer Networks*, 52:2532–2542, 2008.

[29] M. Aykut Yigitel, Ozlem Durmaz Incel, and Cem Ersoy. Design and implementation of a QoS-aware MAC protocol for Wireless Multimedia Sensor Networks. *Computer Communications*, 34(16):1991–2001, 2011.

[30] Injong Rhee, Ajit Warrier, Mahesh Aia, and Jeongki Min. Z-MAC: a Hybrid MAC for Wireless Sensor Networks. *IEEE/ACM Transactions on Networking*, 16(3):511–524, 2008.

[31] Injong Rhee, Ajit Warrier, Jeongki Min, and Lisong Xu. DRAND: distributed randomized TDMA scheduling for wireless ad-hoc networks. In *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '06, pages 190–201, New York, NY, USA, 2006.

[32] Ines Slama, Bharat Shrestha, Badii Jouaber, and Djamal Zeghlache. A Hybrid MAC with Prioritization for Wireless Sensor Networks. In *33rd IEEE Conference on Local Computer Networks (LCN 2008)*, 2008.

[33] V. Raghunathan, S. Ganeriwal, and M. Srivastava. Emerging techniques for long lived wireless sensor networks. *IEEE Communications Magazine*, 44(4):108–114, 2006.

[34] M.D. Jovanovic, G.L. Djordjevic, G.S. Nikolic, and B.D. Petrovic. Multi-channel Media Access Control for Wireless Sensor Networks: A survey. In *Telecommunication in Modern Satellite Cable and Broadcasting Services (TELSIKS), 2011 10th International Conference on*, volume 2, pages 741–744, 2011.

[35] Marion Souil, Tifenn Rault, and Abdelmadjid Bouabdallah. A New Adaptive MAC Protocol with QoS support for Heterogeneous Wireless Sensor Networks. In *17th IEEE Symposium on Computers and Communications (IEEE ISCC 2012)*, Cappadocia, Turkey, July 2012.

[36] C. Buratti and R. Verdone. Performance Analysis of IEEE 802.15.4 Non Beacon-Enabled Mode. *IEEE Transactions on Vehicular Technology*, 58(7):3480–3493, sept. 2009.

[37] Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (Phy) Specifications for Low-Rate Wireless Personl Area Networks (LR-WPANS), 2006.

[38] Shuiming Ye, Mourad Ouaret, Frederic Dufaux, Michael Ansorge, and Touradj Ebrahimi. Error resiliency of distributed video coding in wireless video communication. In *Proc. SPIE 7073, Applications of Digital Image Processing XXXI*, 2008.

[39] M. Souil. Installation steps of the Imote2 platform programming environment, 2010. URL https://www.hds.utc.fr/~marsouil/dokuwiki/doku.php?id=fr:installation.