Wireless Sensor Networks for Pervasive Health Applications

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Eindhoven, op gezag van de
rector magnificus, prof.dr.ir. C.J. van Duijn, voor een
commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen
op woensdag 19 juni 2013 om 16.00 uur

door

Majid Nabi Najafabadi

geboren te Najafabad, Iran

Dit proefschrift is goedgekeurd door de promotor:

prof.dr.ir T. Basten

Copromotor:
dr.ir. M.C.W. Geilen

CIP-DATA LIBRARY TECHNISCHE UNIVERSITEIT EINDHOVEN

Nabi Najafabadi, Majid

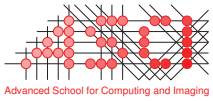
Wireless Sensor Networks for Pervasive Health Applications
/ by Majid Nabi Najafabadi. - Eindhoven : Technische Universiteit Eindhoven, 2013.
A catalogue record is available from the Eindhoven University of Technology Library
ISBN: 978-90-386-3392-3
NUR 986
Subject headings: wireless sensor networks / health application /
wireless body area networks / embedded systems.

Wireless Sensor Networks for Pervasive Health Applications

Committee:

prof.dr.ir T. Basten (promotor, TU Eindhoven)
dr.ir. M.C.W. Geilen (copromotor, TU Eindhoven)
prof.dr.ir. R.H.J.M Otten (Chairman, TU Eindhoven)
prof.dr.ir. J.P.M.G Linnartz (TU Eindhoven)
prof.dr. A. Liotta (TU Eindhoven)
prof.dr. K.G. Langendoen (TU Delft)
prof.dr. G.-Z. Yang (Imperial College London, London, UK)

This work was supported by the Dutch innovation program Point-One through project ALwEN.



This work was carried out in the ASCI graduate school.
ASCI dissertation series number 279.

Printing: Printservice Technische Universiteit Eindhoven

Abstract

Wireless Sensor Networks for Pervasive Health Applications

Wireless sensor networks (WSNs) enable many new applications. Sensor nodes are developing into tiny, light and inexpensive devices, paving the road for widespread use. Health is one of the domains that can greatly benefit from WSNs. Advances in ultra-low power tiny wireless (bio-)sensor devices allow deployment on (and in) a human body during daily life. This forms a Wireless Body Area Network (WBAN) to measure, process, and transmit a variety of physical and physiological data. The combination of WBANs and ambient WSNs provides an infrastructure for pervasive health applications such as elderly care, remote healthcare monitoring of patients with chronic diseases such as COPD, and post-surgery recovery monitoring. Such technology enables patients to live more independently and still receive sufficient care. It also reduces the cost of medical care.

Exploiting well-known general-purpose protocol stacks in WSNs benefits from maturity and technology availability, but suffers from being sub-optimal for specific applications. The protocols need to be adapted to specifications of the target application to fulfill the application requirements. Mobility of individual nodes and of clusters of nodes (WBANs), heterogeneity of sensing characteristics and of quality-of-service (QoS) requirements, and dynamics in the network and its context are special characteristics of pervasive health application. Energy consumption of nodes should be minimized to extend the life-time of the networks and to reduce maintenance cost.

This thesis makes the following contributions. An application-aware Medium Access Control (MAC) protocol, a data prioritization mechanism for data dissemination, and an adaptive intra-WBAN communication protocol are proposed, all aiming to prepare communication protocols for pervasive health applications. The performance of the protocols is studied through real-world experiments with a large-scale ambient WSN and several persons with WBANs. The experiments are complemented with computer simulations to evaluate the protocols in a wider range of use-cases. Models, tools and methods needed to assess performance of the proposed protocols, including a mobility model for WBANs, are developed.

We consider WSNs that exhibit high mobility in the form of moving clusters of nodes (WBANs). The MAC layer needs to efficiently manage medium access in the presence

of cluster mobility in the network. This thesis proposes a TDMA-based MAC protocol called Mobile Cluster MAC (MCMAC). A hybrid schedule-based and contention-based approach is exploited to access the shared medium. Moreover, an optimization mechanism reduces idle listening of static nodes to mobile clusters. Using this mechanism, nodes schedule their listening activities according to a prediction of the presence of WBANs in their communication range. This leads to a considerable energy saving because it substantially decreases unnecessary listening activities of the nodes, without negative influence on the other performance metrics.

The application domain shows a high level of spatial and temporal heterogeneity in terms of sensing characteristics and quality-of-service (QoS) requirements. There is considerable diversity in importance and delivery requirements for different data sources. Without data prioritization, unnecessary service is provided for data with less importance while limiting sources with high demands. This is addressed with a dynamic data prioritization mechanism to disseminate sensed data. It supports varying requirements over time, which allows multi-scenario applications, such as applications with normal and emergency scenarios. The data prioritization mechanism is integrated in a gossip-based data propagation mechanism and realizes robust, fair, and reliable data delivery. The experimental results show that using the prioritization mechanism, the data items with more stringent QoS demands receive better service at the cost of less but sufficient service for less demanding data.

Communication within a WBAN is also very challenging. High mobility due to posture changes, stringent resource constraints due to on-body deployment, and low and time-variant quality of links due to interference with the body are characteristic properties of WBANs. We propose a robust protocol for intra-WBAN communication considering these characteristics. An on-demand listening and data forwarding mechanism automatically adapts the network topology to the connectivity status of the network. It prevents long outage of nodes while optimizing energy consumption. Experimental results show that this mechanism makes a good trade-off between energy consumption and data delivery ratio.

As a prerequisite for performance evaluation of the proposed protocols, we need a WBAN mobility model. A mobility model simulates the relative position of WBANs in the network changing over time. This, in turn, influences the model of the radio connection between nodes. This thesis develops a configurable mobility model, MoBAN, for evaluating intra- and extra-WBAN communication solutions. It models different body postures and individual node mobility; it supports modeling spatial and temporal correlations of WBAN movements and posture patterns. By setting various configuration parameters, the model is adaptable to a broad range of applications, from communication within a single WBAN to applications having a number of WBANs moving in an ambient network. MoBAN is used for simulating the proposed mechanisms in this thesis.

Contents

Acknowledgements

Around five years ago, I moved to the Netherlands together with my wife and my son to continue my PhD education. Such a transition had its own troubles and difficulties. Overcoming those issues was partly possible by my enthusiasm in continuing my studies, but for a great deal, I owe this to the kind and tolerant people who created a wonderful living and working environment around me. This thesis would not have been possible without the support and guidance of many individuals who in one way or another contributed their valuable assistance. I devote this understated acknowledgement to thanking all those who helped me start this path and finish it.

First of all, I would like to express my deepest and sincere gratitude to Twan Basten, for giving me the opportunity to work under his supervision. I am very grateful for his guidance, discussions, and constructive comments on my manuscripts. His support has not been limited only to my research and the thesis; he supported me at all stages of my time in the Netherlands also for things not related to work. His behavior made me feel comfortable to discuss any issue with him to find a solution to overcome the situation. I feel very lucky that I have had such a noble person to work with. Thank you Twan for all your help, patience, and understanding!

Twan was also so kind to offer me a post-doctoral position in the Electronic Systems group to stay in Eindhoven and continue my research and teaching activities. I highly appreciate his effort and I am looking forward to additional fruitful cooperations with him and the other members of the group. For all that, I wholeheartedly thank him.

Marc Geilen was the co-supervisor of my PhD thesis. He contributed in the detail of my work and always came with invaluable, precise, and insightful comments to improve my work and to guide me finding my way. His feedback has been crucial in all steps of my research. I learned lots of valuable points in approaching technical problems and also in truly analyzing and presenting experimental data and results. He has always been there when I needed to discuss something. Without his help, I could not have been at this point and hence, I express my best thanks to him for his help and supervision.

The members of my thesis committee are gratefully acknowledged for reading the concept thesis, providing useful comments, and being present in my defense session. It is my privilege to have Prof. Koen Langendoen, Prof. Jean-Paul Linnartz, and Prof. Guang-

Zhong Yang in the reading committee and Prof. Antonio Liotta in the defense opposition. I would like to thank Prof. Ralph Otten for being the chairman for my defence and also for the flexible environment that he created in the Electronic Systems group.

During my PhD, we had regular meetings to discuss our ongoing research. These discussions were an efficient way to improve our work, think about new research problems, initiate cooperation between the members of the team, and to learn from the work of others. Such discussions sparked many ideas and directed my research to its current form. I gratefully thank Twan Basten, Marc Geilen, Milos Blagojevic, Marcel Steine, Sander Stuijk, Teun Hendriks, Rehan Afzal, and Francesco Comaschi for their contributions in the meetings and discussions.

The great working atmosphere in the Electronic Systems group is certainly never forgotten. I express my best thanks to Morteza Damavandpeyma, Erkan Diken, Shakith Fernando, Sahar Foroutan, Raymond Frijns, Yifan He, Sebastian Moreno Londono, HamidReza Pourshaghaghi, Ahsan Shabbir, Yang Yang, and all other members of the Electronic Systems group for being so friendly, and organizing enjoyable meetings, social events, and coffee breaks. The secretaries of the group, Marja de Mol and Rian van Gaalen, have always been very helpful and hospitable. I greatly thank them for all their help with administrative issues and making the group atmosphere even more friendly and enjoyable. My special thanks go to Rian for her enthusiasm to practice the Dutch language with us.

I express my best thanks to the Iranian families Abedifar, Attarian, Fatemi, Heidari, Mahdipour, Mardanpour, Mousavi, Nikbakht, Pourshaghaghi, Rezaei, Rezaeian, Saeidi, Sheikhattar, Shojaei, Tahvili, and Vahedi for their support and for the great time we had with them. Hamid Shojaei and his wife were very kind and helpful from the first day of our arrival in Eindhoven. Our time in the Netherlands would not have been enjoyable without the wonderful time that we had in particular with families Mousavi, Sheikhattar, Abedifar, and Rezaei. Thank you all our good friends!

Last but definitely not least comes to my family. I am grateful beyond expression to my dearest parents. Words cannot express the extent to which I feel indebted and grateful to them for all their unconditional help and support throughout my whole life. My special appreciations go to my wife Vajiheh, my son Sepehr, and my daughter Sara for their tremendous patience and support. Vajiheh was so kind to leave her family, her job, and her friends to move with me to the Netherlands. She and our children suffered to provide me the possibility to pursue and finish my education conveniently. *I dedicate this thesis to them, with love and gratitude.*

Majid Nabi
May 2013

Chapter 1

Introduction

1.1 Wireless Sensor Networking Technology

Revolutionary advances in integrated circuits, embedded processors, and communication systems in the last decades have resulted in the emergence of new technologies, providing new solutions for the increasing demands of our new connected world. Wireless Sensor Networks (WSNs) are networks composed of numerous tiny sensor devices equipped with wireless radio communication and processing facilities. A single device is called a *sensor node* and can be embedded in various objects or even implanted in living organisms. These nodes communicate and cooperate with each other to form distributed systems, which acquire sensor data and process it to perform some desired tasks. Moreover, actuators can also take part in such networks to enable the system to control specific processes. Such networks are sometimes referred to as Wireless Sensor and Actuator Networks.

Wireless communication and battery powered operation or energy-harvesting make these networks relatively easy to deploy and gives the sensor nodes the freedom to move. These features together with the processing, communication, and data acquisition capabilities make WSNs usable in various domains and applications. Health care [100], fire detection [45], disaster management [30], structural and environmental monitoring [36], wildlife tracking [42], urban supervision and automation, military surveillance [33], and precision agriculture [80] are some applications of WSNs. As just one example out of many, nodes can be distributed over the entire affected residential area after an earthquake to detect alive victims in the collapsed buildings. These nodes autonomously form a network and start sensing and data propagation. Nodes are equipped with very fine-grained sound, temperature, and vibration sensors to detect any sign of life at the disaster site.

Sensor devices are designed according to the requirements of the target application. Sensor nodes are typically supposed to be developed into tiny and low-cost nodes. The small size and low expense of the nodes are prerequisites to using WSNs for many ap-

plications and make the large-scale deployment of these node feasible and justifiable. However, these charming specifications cause some fundamental limitations for sensor nodes. The main issue is the battery capacity as the energy source of the nodes. The amount of energy consumption should be carefully minimized to provide a reasonable life time for the nodes. In many applications, changing batteries is either not possible or very costly. Even in applications with the possibility of easy replacement (e.g. an ambient sensor network in a building), frequent battery depletion becomes annoying and costly due to large-scale deployment. Assuming the average life time of each node in a building with 400 nodes is one year; still every day one node dies, on average, that should be located and replaced.

Considering the size and cost budget and limited energy source of nodes, low power embedded processors or micro-controllers are used as computation units, giving limited computation and storage capabilities compared to advanced high-speed multi-core processor systems. Thus the amount of processing tasks assigned to a sensor node should be limited. Also using very low power radio transceivers limits the transmission range of the nodes. Therefore, nodes are not able to reach far away central stations. That is why a multi-hop data dissemination strategy is mostly used in WSNs. In such cases, each node only has to reach a limited subset of the nodes in the networks (its neighboring nodes).

Regardless of the limitations of the nodes in WSNs, applications have their own requirements. They require a robust and secure system performing their desired tasks with a low deployment and maintenance cost. From the WSN designer's point of view, these demands mean fast and reliable data acquisition with certain throughput by using a number of low-cost sensor nodes providing a very long life time. The application specifies certain Quality-of-Service (QoS) constraints that should be met during the network operation. There is a lot of research on the physical design of nodes and their network protocols (e.g., low-power efficient MAC layers) to satisfy such demands. Typically, a protocol is designed for a specific category of applications to meet its specific constraints.

1.2 Health Applications

In the last decade, an increasing interest has been observed for using WSN technology for pervasive human health applications. By health applications, we envision a broad range of applications in which remote monitoring and supervision are performed. The goals are to improve the quality of life, increase safety, realize independent living, earlier detection of health problems, faster disease diagnosis, and generally to decrease the costs of medical care.

In many health applications, several wireless sensor devices are deployed on or in a human body to sense vital biological signals of the body. These sensors then communicate forming a Wireless Body Area Network (WBAN) [103]. Such on-body network of sensors offers a potential improvement to the person's life. First of all, it relaxes the user from many wired nodes in clinical applications. Moreover, it makes on-line pervasive health monitoring of people feasible. Advances in design and production of very small wireless sensor devices make it feasible and more convenient to wear a WBAN during normal

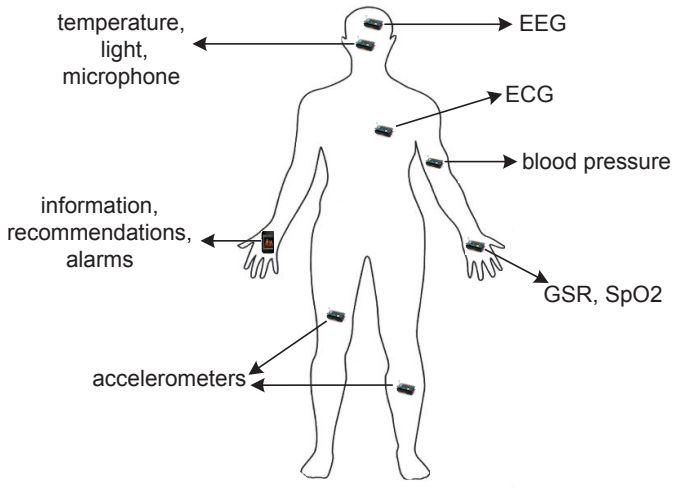


Figure 1.1: Typical body sensors in a WBAN.

daily life. There has been a lot of activity to provide body sensor devices. For instance, specific wearable sensors have been developed in the CodeBlue [99] project at Harvard University to measure several body signals. The Ubimon group at Imperial College has developed custom wireless nodes [102] with an interface to wearable and implantable sensors.

In a WBAN, the sensors are placed at several positions on the body based on the sensor type to measure different biological signals. Fig. 1.1 demonstrates some typical sensor nodes in a WBAN deployment. A subset of these sensors may be selected based on the application and the conditions of the target user. A temperature sensor (for fever), light sensor, and also a microphone may be installed on the head. ECG (Electrocardiogram) sensors may be mounted on several positions of the body to sense the heart signals. EEG (Electroencephalogram) sensors can be mounted on the head to monitor the brain activities through sensing its very weak signals. A blood pressure sensor can be installed on the arms, and an oxygen saturation (SpO2) sensor may be installed on the hands. A GSR (galvanic skin response) sensor may also be mounted on the hand to monitor the emotional state of the patient. Accelerometers are used to measure the movement as well as the tilting of the body.

The sensor nodes in typical WBANs form a heterogeneous mix with a broad range of sampling (sensing) periods and requirements. The corresponding sensor node can transmit the original sensed data in full or perform some preprocessing and transmit the processed data. Only in the latter case, the transmission data rate is often lower than the real sampling rate, which leads to saving radio bandwidth and energy. Such decision is made according to the application requirements. For instance, a sensor node equipped

with an ECG sensor may transmit all sensed data of around 250 samples per second to be processed or recorded by a central station. In another scenario, only the heart beat rate might be enough. So the sensor node needs to perform some processing, calculate the heart rate, and transmit it once per few seconds.

In some applications, a WBAN is used in isolation to gather specific parameters of the body. In other applications, WBANs communicate with a peripheral network with a certain connection hierarchy. Several WBANs together with the surrounding network form a network infrastructure for pervasive monitoring. There are many interesting application areas of WBANs in health applications. We mention some of them, as examples.

Elderly care: WBANs together with an ambient sensor network are able to form a safe environment for elderly people to support them in their daily lives, while being monitored by care workers. Continuous monitoring of elderly people is getting in demand as this group is increasing in number. It includes health monitoring in an elderly center as well as independent living. Their health condition can be estimated from their heart beat rate, blood pressure, and accelerometer data. Strong changes in these parameters, or deviations from the normal range are taken as an emergency situation. An ambient sensor network can sense and control the parameters of the living environment and deliver the body data to a central station. The system may be connected to a health care center for observation and emergency assistance. Useful information such as medication schedules, activity suggestions, and alarms can also be sent to the users/clients.

Daily-life activity monitoring: The activity during daily life of patients with some specific diseases is very important. Patients suffering from COPD (Chronic Obstructive Pulmonary Disease), for instance, have a limitation of the flow of air to and from their lungs, causing shortness of breath. The amount of daily activity should be carefully managed and distributed over the whole day. A static network together with bio-sensors on the body can make a platform for recognizing the patient's activity and can provide useful feedback for him/her. Besides the bio-sensors, other static sensors such as infra-red sensors, pressure sensors on the couches and beds, current sensors for home appliances like a vacuum cleaner can help in a reliable detection of the activities of the patient at home. Such an approach raises much less privacy concerns for patients than using supervisory video cameras or microphones.

In-hospital monitoring: In some cases, patients have to stay in a hospital for intensive care and observations, sometimes for a prolonged period. Meanwhile, several medical parameters are continuously recorded. For a cardiogram, a number of sensors should be attached to the patient. Blood pressure and oxygen saturation may also be of interest to study the patient. When the patient is at risk of sudden faint, a fall detection system is added to the growing collection of sensors. Attaching this number of sensors to a patient with wired connections is very uncomfortable. Instead, using wireless technology and forming a WBAN relaxes the patient and provides freedom of movement. The hospital building can be equipped with static nodes so that the patients wearing the WBAN can freely walk around and stay connected and under proper observation.

Post-surgery in-home recovery monitoring: After a heart operation, the patient will be kept in the hospital for a recovery period. Soon after she/he is sent home, but with a regular schedule of visits by doctors for doing periodic measurements of heart activity and other physiological parameters. A WBAN can help a lot here to make it safer, more convenient, and cheaper. More importantly, such technology provides continuous measurements of the physiological parameters instead of the periodic investigations by the doctors. It can better reveal heart failures and faster detect emergency situations.

Epilepsy diagnostic monitoring: Epilepsy is a prevalent chronic brain disorder with seizures as the main characteristic. About 50 million people around the world have epilepsy [101]. There are diverse types of epilepsy; accurate differentiation between them is especially important to determine the appropriate treatment. Analyzing the EEG signals during a seizure helps to detect the epilepsy type. Seizures are unpredictable and may happen only once in a few months. To catch one, a continuous EEG record is necessary. A WBAN can be a promising solution to provide continuous observations while the patient has his/her own normal life.

Sports training: Precise monitoring of athletes is a modern means to improve their development. Using sensors, all movements of an athlete are recorded and the performance of individual muscles can be measured. Such information is analyzed and used in training. However, presenting normal performance is not possible when a wired set of sensors is attached to the body. Wearable tiny wireless sensors make it more convenient allowing the sporter to perform more realistic movements while equipped with the sensors.

Compared to typical WSNs, sensor nodes in WBANs have tighter size, weight, and power constraints in order to develop into comfortable, wearable, or even implantable devices. Moreover, the characteristics of the wireless channel are different and links are in general of lower quality and more time-variant. On the other hand, health applications usually have tighter QoS requirements because of the importance of human health and the need for early problem detection. These are the points that should be considered in designing a proper WSN architecture for health applications.

1.3 Application-dependent Protocol Design

Many communication protocols have been designed and presented for different networking layers in WSNs. Several Medium Access Control (MAC) protocols have been proposed, which are commonly used and referred to in the community. Data propagation and routing in WSNs have also been studied extensively. Exploiting such general-purpose well-known protocol stacks benefits from maturity and technology availability. The implementations of these protocols are available for common hardware platforms. After years of exploiting these protocols, new versions with improvements have been released. However, using these protocols as they are suffers from being sub-optimal for a particular application at hand. The protocols need to be adapted to specifications and requirements of the target application to fulfill the expectations. In some cases, hybrid protocols com-

bining several existing approaches help to solve the problems of a specific application scenario. There are several factors that should be taken into consideration in making decisions, designing, and configuring a proper protocol stack for a given application. In the following, some of these factors are discussed.

Network size and density: The number of nodes in a network plays a role in making decisions about a protocol. For networks that are expected to grow to very large scales, the protocol should scale well. For small-size networks, the scalability requirement can be relaxed. The physical distribution of the nodes and the size of the deployment area are also important. The connectivity and utilization of the shared wireless channel heavily depend on the node density. In particular, the MAC paradigm being used is determined by the node density. For instance, contention-based MAC mechanisms show good performance in sparse networks while contention-free schedule-based approaches perform much better in dense networks. Network size and node distribution also have impact on the other communication layers such as data routing.

Node mobility: The nodes in WSN applications can show very different movement patterns. In some applications, such as precision agriculture, all nodes are static. In other applications, some sort of mobility may be expected. The type of mobility patterns, the speed, and the movement likelihood are all factors that affect the performance of the network. Moreover, nodes may show only singular mobility or group mobility. Nodes can move very locally or show very global movements across the entire deployment area. A proper protocol stack should be able to manage the expected kind of mobility in the application. Some movements may completely change the network topology and node connectivity. In such a case, the MAC layer and the routing protocol should be very flexible without any strong assumption about the location of the nodes and their connectivity. Instead, in some applications, the movements might be infrequent and manageable with local adaptations. For instance, routing structure adaptation and reconstruction is a means that some routing protocols use for infrequent mobility of sensor nodes.

Sensing specification and traffic load: The sampling rate and resolution of the sensor nodes in the network have a direct impact on the amount of data load in the network. Sometimes data sampling is event-based and a new data packet is generated only in the case of an event. The selected protocol stack should be able to handle the data traffic taking the sampling specifications into consideration together with the network size and topology. These factors influence the decision about the features of all networking layers from physical to data dissemination layer. First the wireless radio features such as the transmission power and data rate should be able to pass the expected traffic. The appropriate MAC and routing protocols are also designed and adapted according to the data load.

QoS requirements: The ultimate goal in designing a WSN is to satisfy the requirements of the running application. Different QoS metrics are usually defined to interpret the application-level requirements into the networking metrics. End-to-end latency and data delivery ratio, and the power consumption of sensor nodes are examples of common QoS

metrics. The application requirements are translated to QoS constraints that need to be met during network operation. Obviously, selecting appropriate networking protocols and devices plays a prominent role in reaching the end goals. Some applications may have a stringent constraint on a specific QoS metric while they can tolerate some variation for other metrics. Protocols used for providing very good latency differ from protocols designed and configured for highly reliable data delivery.

Environment: The deployment area is another determining factor in protocol design for an application. The environment has a big impact on propagation of RF waves. The interference level, RF signal attenuation due to fading and shadowing, and multi-path effects are all environment-dependent factors that affect the data exchange between sensor nodes. Experiences show that outdoor deployments show different behavior than indoor deployments, in terms of the level of interference, multi-path effects, and quality of the wireless links. Moreover, even indoor deployments in different environments might be very different. An industrial environment including many production machines is expected to be more noisy with all kinds of interferences. As another example, the human body strongly affects wave propagation. Thus, WSN deployments in crowded environments suffer from high signal attenuation and packet loss. A protocol designed for communication in an interfering environment should be fortified with proper mechanisms to combat the packet losses. Stronger error detection or correction methods are required. Furthermore, retransmission or acknowledgment methods may be used to provide more reliable data exchange.

Node features: In many applications, there are constraints on the characteristics of the wireless sensor devices such as their size, weight, and cost. Such constraints cause limitations for sensor nodes. A small size requirement means a smaller antenna, which leads to shorter radio transmission range. The energy budget of the nodes (battery) is also constrained and requires very low power devices and protocols. In some applications, even no battery can be used and a power scavenging method is used to provide energy for the sensing, processing, and communication elements of the node.

Heterogeneity and dynamics: Some amount of spatial and temporal heterogeneity may exist in the network. All previously mentioned factors may be non-uniform across the deployment area. Moreover, all parameters may change over time during network operation. The level of heterogeneity is strongly dependent on the application. In some applications, all sensor nodes may be exactly the same with equal sampling specifications. On the other hand, there are many applications in which using different kinds of sensors and wireless devices is unavoidable. In multi-scenario applications, the task being run on the network may even change over time. The protocol stack is designed in such a way as to support the heterogeneity and dynamics expected in the application.

Considering such differences between various WSN applications, the need for application-dependent protocol design and configuration is clear. This is not just to meet the requirements of the application, but also to optimize the network, avoiding unnecessary services which waste resources. Pervasive health, in particular, has char-

acteristic specifications differentiating it from other application domains, which requires specific attention for design, adaptation, and configuration of a WSN protocol stack. Different sensor types, static and body sensor nodes, mobility of individual nodes and of clusters of nodes (WBANs), heterogeneity of sensing specifications and of QoS requirements, and dynamics in the network and its context are important characteristics of pervasive health applications.

1.4 Paradigms for WBAN Communication

In networks with one or more WBANs, biological data sensed by body sensor nodes needs to be delivered to some stations for required analysis and/or storage. Two approaches can be exploited as communication architecture for providing such data delivery.

In the first approach, a *flat architecture* is considered in which the wireless body sensor nodes directly communicate with a peripheral network, a static ambient sensor network, for instance. Using this approach, every body node communicates independently from other body nodes. However, the prerequisite for taking this approach is that all body devices should be equipped with powerful enough radio to reach the surrounding static nodes. In the networks with a dense static network in place over the entire network coverage area, this flat communication architecture of body sensors is a promising solution. Observations in our experimental setup for this thesis show that some body sensor nodes can even better reach the static nodes than the other nodes within their own WBAN. This is because of severe influence of the human body on the propagation of RF waves.

In some health application scenarios, there are limitations for applying the flat architecture for communication by body nodes. A *gateway-based architecture* is the common alternative in this case. A gateway is a node that is also deployed on the body and may have a more powerful radio transceiver with higher transmission range, a more powerful processor for computations, and/or higher storage capacity than other body nodes. In this approach, data sensed by different body sensor nodes is delivered to the gateway node. The gateway node then forwards the received data to a peripheral network. Consider scenarios in which the patient wearing the WBAN can walk outside where no dense wireless static network covers the outdoor area. Here, the body data may be delivered to the person's cell phone [41] that plays the role of a gateway. Then this smart phone (possibly internet-enabled) transmits the data to a certain base station using the cell phone communication infrastructure. Note that fortifying all body nodes with cell phone communication facilities is typically not an applicable approach. In scenarios in which none of the body sensor nodes has cell phone communication features, the body data might be stored in the memory of the gateway when the person is not in an area covered by a static network. When the person goes to such a covered place (his/her home, for instance), all stored data may be uploaded to the static network by the gateway node. In some health applications, ultra-low power body nodes [92] may be used whose radio range barely reaches one meter. In such case, even if the area is covered by a static network, it is more applicable to use the gateway-based approach to relay body data to the peripheral network.

There are also other scenarios that motivate using a gateway-based communication architecture. In some applications, we need to have a gateway in the WBAN not only for forwarding body data, but also for information processing and data aggregation [76]. In such a scenario, the gateway may collect data from various sensors on the body and process it to find out the health status of the person. Transmitting more abstract information about the patient health can substantially decrease the network load compared to transmitting the whole fine-grained sensed data to the base stations.

1.5 Contributions

This thesis proposes mechanisms to provide an optimized WSN protocol for health monitoring applications. It includes combining various communication paradigms and adapting the protocols in different networking layers toward the requirements of these applications. A MAC mechanism, a priority-based data dissemination mechanism, and an adaptive intra-WBAN protocol are proposed. In each step, the performance of the protocols is studied through real-world experiments with a large-scale ambient WSN and several persons with WBANs. The experiments are complemented with computer simulations to evaluate the protocols in a wider range of circumstances. On the road toward such goals, the required models and frameworks for performance evaluation of the proposed protocols are developed.

The MAC layer is responsible for efficiently managing the access of different radio transmitters to the shared channel. As an important aspect, the MAC protocol should provide appropriate support for the node mobility in the network. Health applications exhibit high mobility behavior in the form of moving clusters of nodes (WBANs), which is called *cluster mobility* in this thesis. When the network exploits a gateway-based architecture for WBAN communication, the mobility may be approached as singular node mobility (the gateway). This kind of mobility is well studied in the literature and several MAC mechanism have been presented for supporting that (e.g., [12, 98]). However, for a flat architecture, we face the mobility of a cluster of independent nodes. Dedicated cluster mobility support in the MAC protocol would be beneficial in a flat architecture to efficiently manage WBAN communications.

This thesis proposes a particular MAC protocol, called Mobile Cluster MAC (MC-MAC), to support cluster mobility. It considers a flat architecture for communication of mobile cluster nodes with the static network. Time Division Multiple Access (TDMA) mechanism is considered as the base for developing the MAC mechanism to exploit the high-efficiency of this paradigm. It then exploits a hybrid schedule-based and contention-based approach to access the shared medium. Experiments and simulations show that using this mechanism considerably reduces collisions and leads to more efficient data delivery in comparison to the alternative mechanisms. Furthermore, an optimization mechanism is developed to schedule listening activities of the nodes to the WBAN nodes. It estimates the presence of the WBANs in the communication range of the node and based on that avoids unnecessary listening to the channel. This leads to considerable energy savings. The experimental results show that this mechanism decreases around 70% of

unnecessary listening to the mobile clusters, on average in the experiments, without a negative influence on the performance of the network.

The health application domain shows a high level of spatial and temporal heterogeneity in terms of sensing characteristics and QoS requirements. There is considerable diversity in importance and delivery requirements for different data sources. For instance, the body data may be more important with tighter delivery requirements than the ambient data. There are even differences in importance of various body data, based on the signal being sensed. Without data prioritization, unnecessary service is provided for data with less importance while limiting the source nodes with higher demands. This is addressed with a dynamic data prioritization mechanism for data dissemination on top of the MCMAC layer. At the same time, it supports requirements varying over time. Multi-scenario health applications such as applications with normal and emergency scenarios have requirements and sampling specifications that vary over time when switching between different scenarios. The data prioritization mechanism is useful for both flat and gateway-based WBAN communication architectures. The mechanism is integrated in an adapted gossip-based data propagation mechanism to realize a robust and fair data delivery mechanism. Several simulations and real-world experiments using various heterogenous deployment setups are performed to observe the behavior of the data prioritization mechanism. The results clearly show that using dynamic data prioritization, the services in the network are distributed in accordance with the QoS requirements of data items from different sensor nodes.

The MCMAC protocol together with the priority-based gossiping data dissemination mechanism realize a platform for receiving body data as well as ambient information and delivering these to the sink nodes. When a gateway-based approach is used for delivering WBAN data to the peripheral network, providing an efficient, reliable, and robust mechanism for communication of the body nodes to the gateway within the WBAN is of great importance. Frequent topology changes due to high mobility and posture changes, more stringent resource constraints due to on-body deployment, and low and time-variant quality of links due to influence of the body are characteristic properties of these networks. A robust mechanism is proposed for intra-WBAN communication. All nodes need to deliver their data to a gateway, also deployed on the body. An on-demand listening and data forwarding mechanism (ODLF) automatically adapts the network topology according to the connectivity status of the body sensor nodes to the gateway node. It prevents long outage of nodes while optimizing their energy consumption, leading to a more robust connection of the body sensor nodes to the gateway. Several experiments are done to observe the behavior of the protocol in real deployments. The results show that this protocol gives a good trade-off in terms of performance metrics and energy consumption and can be a generic solution for communication in many WBAN application scenarios. Experiments with various configurations of nodes show that the protocol properly adapts itself to the dynamics in the WBAN.

In different parts of the thesis, a WBAN mobility model is needed as a prerequisite for performance evaluation of the proposed protocols. A mobility model determines the relative position of WBANs in the network at any time during simulation. Thus, it affects the radio connection between nodes. This thesis develops a configurable mobility model,

MoBAN, for evaluating intra- and extra-WBAN communication solutions. It models different body postures and individual node mobility and it supports modeling spatial and temporal correlations. The model is adaptable to a broad range of applications, from communication within a single WBAN to applications having different number of WBANs moving in an ambient static network. This mobility model is used for simulating the proposed protocols in this thesis. It is also adopted in the MiXiM [47] framework on top of the OMNeT++ 4 [5] network simulator and is integrated in MiXiM 2.1 release.

1.6 Thesis Overview

The thesis continues in Chapter 2 with a description of the intended network architecture and preliminaries that are used throughout the entire thesis, such as the definition of the QoS metrics and the performance evaluation methods. The mobility model for body area networks, MoBAN, is explained in Chapter 3. It introduces the mobility behavior that is expected in the intended health applications as well as the implementation of a configurable mobility model simulator for computer simulations of networks including WBANs. Chapter 3 is based on publication [60].

Chapter 4 describes the MCMAC protocol. First, it reviews the prominent MAC protocols used for communication in WSNs, focusing on their method for handling node mobility. Subsequently, it presents the hybrid approach for accessing the wireless channel by the static and body sensor nodes. It investigates two different contention-based paradigms to be used in a TDMA-based communication infrastructure, and analyzes their respective performance. It also provides guidelines for selecting a proper approach based on the features of the network and hardware setup at hand. This chapter then presents the listening scheduling method for avoiding idle listening to the mobile nodes in the network. Experimental evaluation results of this MAC layer are explained in detail. Chapter 4 is based on publication [59].

The proposed dynamic data prioritization mechanism for data dissemination is presented in Chapter 5. The chapter starts with an introduction to different heterogeneity sources of a WSN and an overview of the existing methods for data prioritization in WSNs. The used gossiping strategy for data propagation towards the sink nodes in the network is then introduced. Subsequently, the dynamic data prioritization mechanism is explained including a QoS requirement statement in the packets and the priority assignment strategy. Afterwards, the experiments and simulations are described that show how the mechanism differentiates between various importance classes in different WSN setups. Chapter 5 is based on publication [58].

Chapter 6 focuses on communications within a WBAN, for the networks that use a gateway-based communication architecture. It first points out the characteristic features of WBANs that differentiate them from typical WSNs, signaling the need for very careful protocol design for intra-WBAN communication. Different protocol design approaches are studied and their respective advantages and drawbacks are discussed. The chapter continues with the proposed on-demand listening and data forwarding mechanism. It explains the kinds of disconnections that are to be avoided and the method for detecting

and overcoming such outages. Several experiments have been done by deploying nodes on a body to assess the performance of the protocol in different network conditions and body postures. This chapter explains the setup, experiments, and the results. Chapter 6 is based on three publications [57, 61, 62].

Chapter 7 concludes the thesis and provides a vision about future research in the field of WSNs for health applications.

Chapter 2

Preliminaries

2.1 Overview

This chapter presents definitions, assumptions, and platforms that are used throughout the rest of the thesis. It first describes the intended network architecture for health applications and continues with the QoS metrics that are considered for evaluating the behavior and performance of the developed protocols in this thesis. Furthermore, it explains the means that are used to evaluate these QoS metrics for a given network setup, including real-world experiments and computer simulations.

2.2 Network Architecture

We decompose the network for a health monitoring application scenario into two parts, the *static network* and *mobile clusters*. The main difference between the nodes in these two parts is their movement. However, there are other important differences such as the type of signals being sensed, communication capabilities, size, and energy constraints.

2.2.1 Static Network

This part of the network consists of the nodes that are installed at various places in a building, including on furniture. The nodes in this network are mostly static with the possibility to be occasionally relocated (limited mobility). This network is also called the *ambient network* in health applications. Suppose that $S = \{s_1, s_2, \cdots\}$ is the set of $N_s = |S|$ static sensor nodes. A small subset $Sinks \subset S$ of the set of static nodes is considered as the set of sink nodes (central stations). Data samples over the entire network are supposed to be collected at the sink nodes. The sink nodes may process the gathered information, store it, and communicate with other networks such as a medical center. The number and location of the sink nodes are determined according to the circumstances in the deployment and considering the QoS requirements [16].

Nodes may have sensors and generate data that is supposed to eventually reach a sink node. Besides this sensing task, the static nodes are responsible for making a multi-hop data dissemination backbone to deliver sensed data to the sink nodes. This includes both the sensed data from static nodes and the data generated by mobile cluster nodes (body sensor nodes).

2.2.2 Mobile Clusters

In a deployment for health applications, there are WBANs, each including a number of sensor nodes. We approach WBANs as clusters of nodes moving together (mobile clusters). Because of their on-body deployment, these nodes show a high mobility in the sense of both group mobility and individual mobility within the cluster. We assume that we have a set of N_{mc} mobile clusters in the network denoted by the set $MC = \{C^1, C^2, \cdots, C^{N_{mc}}\}$, where $C^i = \{c_1^i, c_2^i, \cdots\},\ 1 \leq i \leq N_{mc}$ is a mobile cluster including $N_m^i = |C^i|$ sensor nodes. The total number of nodes in the network (network size N) is then given by Eqn. 2.1.

$$N = N_s + \sum_{i=1}^{N_{mc}} N_m^i \tag{2.1}$$

The sensor nodes in a mobile cluster are deployed on different positions of the body and sense various biological parameters. The sensed data needs to be delivered to the static nodes around to be further disseminated toward the sink nodes. In the rest of the thesis, we use the notation η to refer to a sensor node whenever it may be any static or mobile cluster node.

2.2.3 Data Sampling and Transmission

Every wireless node in the network can be equipped with sensors to sense particular parameters. Such nodes are sources of data in the network. A *data item* is considered to be one sample of the parameter being sensed. Depending on the application scenario, sampling can be *event-based* or *periodic*. In event-based sampling, the node generates a data item whenever a certain physical event is detected. For instance, an infrared sensor may generate a data item whenever an object (person) passes. In periodic sampling, node η samples the signal in equally spaced time intervals $T_s(\eta)$, where node η may be any static or mobile cluster node. A new data item is generated every $T_s(\eta)$ units of time, which is called the *sampling period*. The sampling period is usually defined by the application requirements and the nature of the signal under observation. The data item sampled by sensor node η at time t is referred to as Γ_η^t.

Every source node needs to transmit the data samples to be eventually delivered to a sink node. A *packet* is what a sensor node transmits on the air in one transmission; its length usually depends on the used wireless radio device. A packet includes control information from different layers such as the MAC layer as well as one or more data items; assume that λ denotes the number of data items in a packet. Node η has a certain

transmission period $T_{tx}(\eta)$. So every $T_{tx}(\eta)$ units of time, the node transmits a data packet that includes $\lambda \geq 1$ sampled data items. The data items included in a packet may be the node's own data sample or the received data from other nodes to be forwarded to the next hop in the data dissemination path. In this thesis, we focus on periodic data sampling and packet transmission.

2.3 Performance Metrics

All WSN applications, including health applications, have specific QoS requirements that are expected to be satisfied by the network deployment. The metrics of interest and their exact definition are dependent on the application scenario and the network architecture. Reliability, latency, life time, and robustness are examples of such QoS requirements. An application may specify certain constraints for its QoS metrics of interest. The network is then expected to provide services in such a way as to meet the given QoS constraints. A WSN is a distributed network of many independent wireless devices. These devices have randomness in their behavior and the network as a whole exhibits a stochastic behavior. Because of that, there is always some chance that certain QoS constraints are violated in some part of the network or at some time during the network operation. Therefore, the constraints typically are soft constraints that can tolerate a certain level of violations.

In this thesis, we develop mechanisms to provide suitable services for health applications. To evaluate the performance of the proposed mechanisms, we consider some low-level metrics that investigate the detailed behavior of the network. These metrics may directly reflect the application-level QoS metrics. However, some others may not directly translate to high level metrics, but allow us to further observe the behavior of the network. This section presents the definition of these metrics and the methods to calculate the value of these metrics in our intended network architecture.

2.3.1 Reliability

Reliability of a network reveals the ability of the network to deliver the sensed data to the sink nodes. Data items may be lost during dissemination toward the sink nodes. Link failures, interference, and collisions are causes of *packet loss*. If the packet losses are not dealt with properly, the sensed data may not reach any sink, resulting in *data loss*. Depending on the routing protocol, earlier data items may be overwritten by the subsequent data items in the relaying nodes when the memory space dedicated for storing data from each sensor node is limited. This is also a source of data losses, specifically in the data dissemination mechanism that is used in this thesis.

To quantify the reliability of a network, Data Delivery Ratio (DDR) is used that computes the percentage of the generated data items by each node in a certain time period that reach a sink node. $\mathcal{R}^t(\eta)$ is defined as the DDR of node η at time t that shows the percentage of the $\mathcal{H}$ last data items sampled by node η before time t that reached a sink, calculated by Eqn. 2.2.

$$\mathcal{R}^t(\eta) = \frac{1}{\mathcal{H}} \sum_{k=1}^{\mathcal{H}} D^{t,k}(\eta) \tag{2.2}$$

where $D^{t,k}(\eta)$ is a 0 or 1 valued parameter that shows whether the k^{th} data item sampled by node η before/at time t (starting to count backward from time t) has reached a sink; a value of 1 indicates a successful delivery. The length of the averaging history $\mathcal{H}$ is an important parameter. A lower value allows us to observe short-term data delivery, while a longer window length provides a long-term sense of the network performance in delivering the sensed data items to the sink nodes.

The application may specify a constraint $\mathcal{RC}(\eta)$ for a DDR of each node η in the network. It means that the application expects the value of $\mathcal{R}^t(\eta)$ to remain above this threshold ($\mathcal{R}^t(\eta) \geq \mathcal{RC}(\eta)$). However, because of the stochastic nature of the network, this may not be 100% satisfiable during the network operation. Therefore, the application needs to have tolerance to a certain percentage of violations of its constraint $\mathcal{RC}(\eta)$. As an example, an application may require the DDR for a node η to be higher than 80% in 95% of the time. Note that the application requirements may be different for different nodes because of the differences in the signals that the sensor nodes are sensing. Moreover, in this thesis, we consider scenarios in which the QoS constraints may change over time. A health monitoring application scenario with normal and emergency situations for WBANs is an example. Tighter constraints may be considered for the nodes in a WBAN when the person wearing it is detected to be in an emergency condition.

The distribution of data losses in the network over time is also important; it can further show the detailed behavior of the network in delivering data to the sink nodes. For performance evaluation of the proposed protocols in this thesis, we also investigate the length of burst data losses. It is defined as the number of subsequent data losses observed during an experiment. For any time t such that $D^{t,1}(\eta) = 1$, $\mathcal{BL}^t(\eta)$ is defined as the length of a burst data loss that happened right before the last successful delivery before time t; it is calculated by Eqn. 2.3.

$$\mathcal{BL}^t(\eta) = l - 2 \tag{2.3}$$

where the l^{th} data item generated before time t is the last received data item before the burst, stated by Eqn. 2.4.

$$D^{t,i}(\eta) = 0 \;\; for \; i = 2, ..., l-1 \quad \wedge \quad D^{t,l}(\eta) = 1 \quad \wedge \quad D^{t,1}(\eta) = 1 \tag{2.4}$$

2.3.2 Latency

It is of importance in many applications how fast the sensed data reaches the sink nodes over the network. The latency of a sampled data item Γ_η^t is the time between the moment of sampling (t) at the source node η until the time it arrives at a sink node. In the case of several sink nodes, the earliest data delivery to a sink is taken as the latency of that

data item. Let $t_a(\Gamma, \eta)$ be the time of the first arrival of data item Γ at some node η. $\mathcal{L}(\Gamma)$ denotes the latency of data item Γ calculated by Eqn. 2.5.

$$\mathcal{L}(\Gamma^t_\eta) = \left(\min_{s \in Sinks} t_a(\Gamma^t_\eta, s) \right) - t \tag{2.5}$$

where $t_a(\Gamma^t_\eta, s) = \infty$ if data item Γ^t_η does not reach sink s. The latency value consists of the time that a data item is traversing the path toward the sink and the time that the data is suspended in the relaying nodes in the routing path due to possible congestion. It is clear that we are not able to calculate a latency value for a lost data item.

An application may require different latency levels for different nodes based on the nature of the signal being sensed. Let $\mathcal{LC}(\eta)$ be the latency constraint required for data items from node η. It means that the application requires the latency of the data items generated by node η to be less than $\mathcal{LC}(\eta)$. As we discussed for DDR, there are always chances for violation of these constraints for some data items because of the stochastic behavior of the network. A common specification is to require that for a certain ratio of the received data items at the sink nodes generated by node η, the achieved latency be lower than the constraint. Moreover, different applications may have different requirements for the data items that arrive at a sink node, but with a violated latency. In one scenario, even if the latency deadline of a data item has expired, the data item still has value and the application is interested to keep its latency as low as possible. In another scenario, the ratio of data items meeting the latency constraint can be of higher importance. In such a scenario, once the latency constraint expired for a data item, it may not be of interest to the application anymore. In this thesis, we use the average latency of the data items from different nodes for evaluating our mechanisms. This evaluation corresponds to the former application scenario.

2.3.3 Data Age

Some application scenarios may need to keep their observations of the environment under supervision as up-to-date as possible. As an example, consider an ambient network in which the application maintains a temperature map of the region under supervision. Here the application is interested to keep a temperature map of the last updates of temperature in different parts of the area. In such cases, none of the latency and DDR metrics can directly show how up-to-date is the gathered information. Such applications are rather interested in the *age* of the gathered sampled data at the sink nodes. The latency in the data delivery to the sink has indeed an impact on the age. Data losses also play a role here. However, computing the age of the gathered data items at the sink node can reflect at once the requirement of the application. Notice that the age metric does not replace latency or DDR metrics.

We consider data age at the sink nodes, besides DDR and latency, to further investigate the various performance aspects of the proposed mechanisms. The age ($Age(\eta, s, t)$) of data items from node η at sink s at time t is the time difference between the current time t and the sampling time t_s of the last received data item generated by node η at sink s, given by Eqn. 2.6.

$$Age(\eta, s, t) = t - \max\{u \mid t_a(\Gamma_\eta^u, s) \leq t\} \tag{2.6}$$

In the case of several sink nodes in the network, the age $Age(\eta, t)$ of node η at time t is the minimum age of that node at all sinks, given by Eqn. 2.7.

$$Age(\eta, t) = \min_{s \in Sinks} Age(\eta, s, t) \tag{2.7}$$

If a data item from node η is lost, the value of $Age(\eta, t)$ increases because the existing received data item from node η gets older at the sink nodes.

2.3.4 Energy Consumption

Sensor nodes in typical WSNs are equipped with constrained energy sources, mostly in the form of small-sized batteries. This limitation together with the desired long lifetime of sensor nodes make minimizing the energy consumption of the nodes a very important issue in WSNs. Health applications including WBANs, in particular, have very tight constraints on energy consumption. Body sensor nodes are either equipped with very small batteries or use energy scavenging, both implying very limited energy resources.

Realizing low power sensor nodes starts with using ultra-low power sensors, processors, and radio devices. However, the communication protocol has a strong impact on the energy consumption of the node. Energy consumption of the radio communications constitutes a substantial part of the overall energy consumption of the node. Thus, the communication protocol is supposed to avoid energy dissipation due to unnecessary radio activities. This is especially a key design objective for MAC protocols.

There are several terms that constitute the total energy consumption of a node. The radio communication energy E_{radio}, processor energy consumption E_{proc}, and sensing energy E_{sense} by different sensors of a node are the main terms. Considering these energy consuming parts, the total energy consumption of a wireless sensor node in a certain period, say one transmission period, is given by Eqn. 2.8.

$$E_{total} = E_{radio} + E_{proc} + E_{sense} \tag{2.8}$$

E_{radio} is the dominant and most variable part of energy consumption in many WSN nodes [91]. It heavily depends on the underlying communication protocol. In the next section, we present a parametric equation for calculating the energy consumption of wireless nodes that we use for performance evaluation of our protocols.

2.4 Performance Evaluation Methods

2.4.1 Experiments and Simulations

Evaluation of different performance metrics for a given WSN deployment setup is an important phase of designing a network protocol. Performance evaluation is typically done using either real experiments or computer simulations. Real experiments test and evaluate the protocols on real wireless nodes in an environment similar to the application environment. A real experiment proves the implementation feasibility of the approach considering computation and communication limitations of real WSN nodes. Moreover, experiments provide more realistic results because simulation models cannot completely and perfectly mimic the real network behavior. Wireless radio, the communication channel, and interference are still difficult to model precisely in simulation. The models cause some evaluation inaccuracy, which leads to inaccurate estimation of the performance metrics. Using a real network deployment, we make sure that the protocol works in a realistic network setup.

Although experiments provide more realistic evaluation, there are limitations on the scale of the network and the number of experiments that can be performed. Thus, simulation is still of great importance because it is less expensive, extensible to larger scale networks, and can be done and repeated in a shorter time. A protocol can easily be evaluated with various scales, node densities, and configurations. It allows investigating the dependencies between the behavior of the protocol and the network size and density, which leads to more comprehensive evaluation of the protocols. Moreover, by doing simulations, we have better controllability and observability over the network setup, configuration, and behavior. It helps us to better observe the behavior of the protocol and detect potential bugs in the design and implementation process. Another issue that motivates us to use computer simulations besides real experiments is mobility in the network. By simulating mobile networks we can evaluate and compare different protocols with exactly the same mobility patterns. It is very difficult or even infeasible to perform different experiments with exactly the same human mobility patterns. Mobility has a great influence on the network topology, node density in different parts of the network, and connectivity. When the mobility patterns are not the same in two experiments (running different protocols or configurations, for instance), having a fair comparison of the achieved performances is hard. In simulations we can generate certain mobility patterns and use it for multiple simulation runs.

In this thesis we use both simulations and real experiments for performance evaluation of all proposed protocols and mechanisms. In general, experiments and simulations that are done in this thesis are of two types. The first kind is intra-WBAN experiments in which communication within a WBAN is evaluated (Chapter 6). The second kind of experiment consists of a static network and several WBANs. This type of experiment evaluates all communications in the network to deliver sensed data to the sink nodes in the static network (Chapter 4 and 5). In the following, the experimental platform and the simulation framework are described.

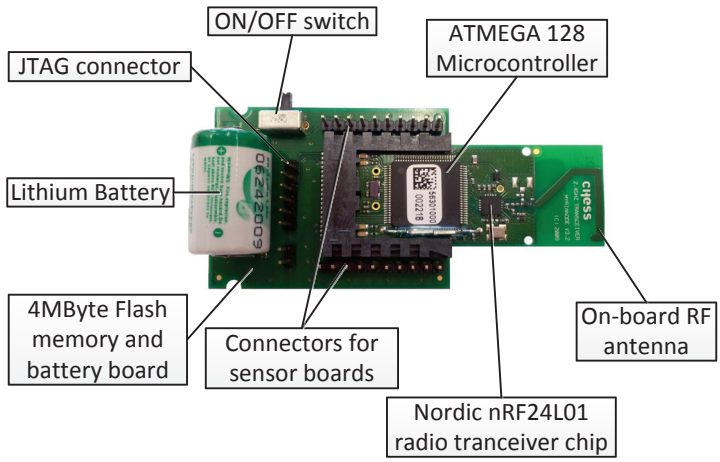


Figure 2.1: A MyriaNed [95] node used in the experiments.

2.4.2 Experimental Platform

We investigate the applicability and performance of every proposed protocol in this thesis through various real-world experiments. We use a testbed of 76 *MyriaNed* [95] wireless nodes (Fig. 2.1) with various deployments and setups considering health monitoring application scenarios. To evaluate the whole network, we deploy a rather large-scale indoor static network on two floors of the Electrical Engineering department of the Eindhoven University of Technology (TU/e), which covers an area of around $2400m^2$. Several people took part by wearing wireless nodes (forming mobile clusters) and by performing their normal behavior during office work. For intra-WBAN experiments, we deploy at least 12 sensor nodes on a body performing various posture patterns to evaluate the WBAN communication protocols in all postures and circumstances. The exact experimental setup for each experiment is presented in the corresponding chapter. In the following, the specifications of the wireless nodes and the available protocol stack are explained.

Hardware Features

Fig. 2.1 shows a MyriaNed node and its different elements. This wireless node features an ATMEGA128 microcontroller and a *Nordic* nRF24L01 radio chip [4] as transceiver. The radio chip works in the 2.4GHz ISM band, which is one of the proposed frequency bands by IEEE 802.15.4 [10] for Wireless Personal Area Networks (WPANs) and IEEE 802.15.6 [11] for WBANs. The radio chip uses a data rate of 2Mbps and packets with a fixed size of 32 bytes. Taking the packet preambles into account, the transmission time of a packet is around $T_{transmit} = 164\mu s$. The radio can be set in receive, transmit,

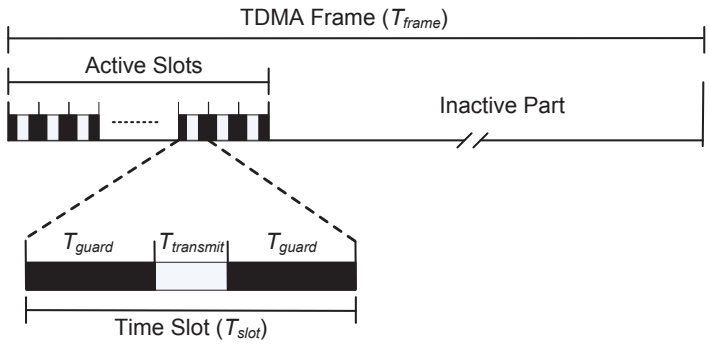


Figure 2.2: The basic structure of the TDMA frames and time slots.

or idle modes. The current of the radio chip in receive mode is $I_{receive}$ =12.3mA. In transmit mode, four transmit power levels (-18, -12, -6, 0dBm) can be set for the radio with currents of $I_{transmit}$ = 7.0, 7.5, 9.0, and 11.3mA, respectively. Using experiments with different transmit power levels allows us to test the protocols in various network conditions and connectivity patterns. The radio consumes very little current in the idle mode ($I_{idle} = 22\mu$A).

Each node is equipped with a 4MByte flash memory used to log data such as the radio activities and other useful information during the experiments. After finishing an experiment, the logged data is downloaded from the nodes and analyzed to extract performance metrics.

Protocol Stack

A TDMA-based MAC protocol, called gMAC [94], is available as the base MAC layer for the MyriaNed nodes. In the TDMA mechanism, the communication activities of the nodes in the network are performed in a periodic manner using fixed-size TDMA frames. A *frame* consists of two parts, an *active* part and an *inactive* part. Fig. 2.2 shows the general structure of a TDMA frame. The active part of the frame includes a fixed number of *time slots* that are used for communications. The channel is silent during the inactive part of the frame because none of the nodes transmits in that part. Therefore, all nodes go to the idle mode in the inactive part of the frame leading to low average energy consumption for nodes. The slots in the active part of the frames are scheduled to different nodes in a neighborhood for their packet transmission. Several slot scheduling mechanisms are currently provided to assign slots to the nodes for transmission. Simple fixed slot assignment and unique slot assignment in two-hop neighborhood, similar to the LMAC [96] slot scheduling strategy, are available. A node transmits its data in its scheduled slot and listens in (some of) the other active slots to receive packets from its neighbors.

Wireless nodes are synchronized using a decentralized frame synchronization mechanism [13]. In every frame, each node computes the phase difference of its TDMA frame with the frames of all its direct neighbors. Based on the measured phase differences, the node adjusts its frame using, for instance, a median [93] algorithm. A guard time is inserted at the beginning and the end of every active slot to avoid problems due to small phase differences (Fig. 2.2). In a transmit slot, relative to its own clock, the node waits for one guard time (T_{guard}) and then starts packet transmission. In receive slots, the receiver node starts listening to the channel from the the beginning of the slot. In the current implementation, the node keeps listening for the entire slot. If the next slot is also a receive slot, the radio is already in receive mode and no mode switching is necessary. An alternative approach is that the radio is set to the idle mode once a packet is received. This can save energy in the slots in which the listening node is in the transmission range of a transmitter node and it receives a packet.

Each TDMA slot of the MAC layer is of length $T_{slot} = 764\mu s$ [1], which consists of a transmission time ($T_{transmit} = 164\mu s$) and two guards ($T_{guard} = 300\mu s$) shown in Fig. 2.2. The TDMA frame length T_{frame} is adjustable (we use one second in most of our experiments). Considering the very small size of the slots and the limited number of active slots in each TDMA frame, nodes use a very low duty cycle, leading to very low average energy consumption. Having 16 active slots in each one second frame, for instance, the duty cycle is around 1.22%.

A transmission packet consists of the MAC header (including the ID of the transmitter) as well as λ data items. Fig. 2.3 illustrates the structure of a transmission packet in the default protocol stack of MyriaNed. A data item includes the sampled data, the ID of the source node of the data sample, the time of sampling, and other control information that may be required. The number of data items per packet depends on the size of the packet, the MAC header size, and the size of each data item. In all our experiments, each transmission packet includes three data items ($\lambda = 3$) of size 8 bytes each.

The gMAC protocol does not include any receiver ID to address a particular receiver for the packet. All listening nodes in the range receive the transmitted packet. It makes the MAC layer suitable for running an epidemic data dissemination protocol. The MyriaNed protocol stack provides a gossiping strategy as proposed in [29]. In this protocol, every node has a local memory space (cache) that can store several data items from other sensor nodes in the network. In every round (TDMA frame), a node randomly selects $\lambda - 1$ data items from its cache and puts them into its transmission packet together with its own most recent sensed data (assuming it has a sensor). It then propagates the packet to its neighbors. Once a data item is received by a node, it is stored into the cache if the item is newer than the existing one, if there is any (older data items are overwritten by newly sampled data items). Thus at any time, at most one data item from any node is stored in the cache.

The gossiping mechanism provides robust data dissemination because of the built-

[1] Here, switching to the desired mode (receive or transmit) in the slots in which the radio is initially in the idle mode takes place during the first guard time. However, in an implementation, a certain part of the slot may be dedicated for mode switching. In that case, the length of the slot will be longer. The latter case is used in our earlier publications.

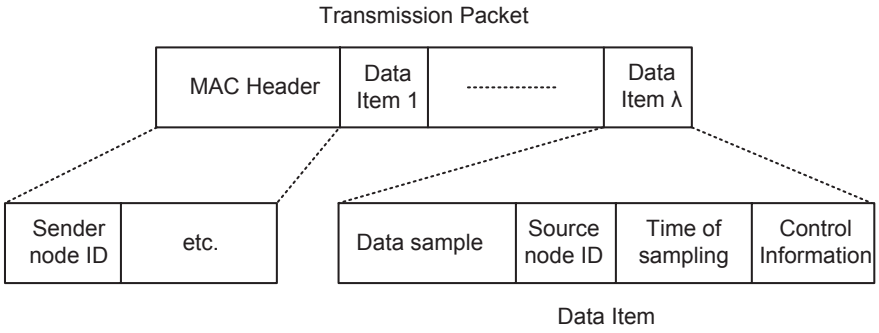


Figure 2.3: A transmission packet in the default protocol stack.

in redundancies in the network. A data item may be stored in the cache of several nodes and may be propagated in different directions. Thus link failures will cause low data losses. With gossiping, the network does not rely on any specific routing path for data dissemination. It relieves the network from creating and maintenance of data dissemination routes. Although such gossiping improves robustness and reduces the overhead of routing structure maintenance, it usually provides higher end-to-end latency. Smarter data forwarding taking the local network status into consideration can improve the overall performance of this store-and-forward gossiping mechanism. An example of such adaptations is presented in [15]. It exploits a probabilistic acknowledgement mechanism with controlled data item retransmission to improve the latency and DDR.

MyriaNed Energy Consumption

The radio energy consumption (E_{radio}) constitutes the dominant part of the total energy consumption of a node. The MyriaNed nodes use a TDMA-based communication mechanism in which the energy consumption is identical for every frame. We calculate radio energy consumption per TDMA frame. Eqn. 2.9 gives the radio energy consumption and consists of energy consumption in transmit, receive, and idle modes.

$$E_{radio} = E_{transmit} + E_{receive} + E_{idle} \tag{2.9}$$

Transmit energy in one transmit slot includes the energy consumption for transition to the transmit mode plus the packet transmission energy. As there is only one transmission slot for each node per frame in the MAC implementation, the transmit energy per frame ($E_{transmit}$) is equal to the energy of transmitting in one slot. Eqn. 2.10 calculates $E_{transmit}$.

$$E_{transmit} = (I_{transmit} \times T_{transmit} + I_{sw} \times T_{sw}) \times V_{bat} \tag{2.10}$$

where $I_{transmit}$ and $I_{sw}(= 8.0\text{mA})$ are average radio current during packet transmission and during switching to the transmit mode, respectively. $T_{transmit}$ and $T_{sw}(= 130\mu s)$

stand for transmission time and radio mode switch time in the radio chip. V_{bat} is the voltage of the battery, which is 2.4V for these nodes. As an example, the transmit energy consumption of MyriaNed per frame when using the highest transmit power level (0dBm) is $E_{transmit} = 7\mu$J.

Receiver energy consumption in each frame depends on the number of slots in which the node listens to the channel (N_{listen}) and is calculated as follows.

$$E_{receive} = [I_{receive} \times (\underbrace{T_{transmit} + 2 \times T_{guard}}_{T_{slot}})] \times V_{bat} \times N_{listen} \tag{2.11}$$

where $I_{receive}$ is the radio chip current in the receive mode. Assuming $N_{listen} = 15$ in an experiment, the receive energy consumption of MyriaNed in one frame will be $E_{receive} = 338\mu$J. Note that in the calculation of the receive energy consumption in Eqn. 2.11, we consider the current implementation in which the node keeps its radio in the receive mode for the whole slot length. Thus it provides a conservative estimation of the receive energy consumption. If we consider an implementation that switches the radio to the idle mode after reception of a packet, we need to take the distribution of the amount of clock phase differences between nodes into account and based on that estimate the energy consumption for the receive slots with a packet reception. However, throughout this thesis, we use the calculation in Eqn. 2.11 based on the current implementation of the TDMA-based MAC layer for MyriaNed nodes.

No radio activity is performed in the inactive part of the frame and the nodes are in the idle mode. Given that the radio current in the idle mode is $I_{idle}(= 22\mu A)$, the energy consumption of a node in the idle mode per frame is calculated by Eqn. 2.12.

$$E_{idle} = I_{idle} \times (T_{frame} - T_{slot} \times N_{active}) \times V_{bat} \tag{2.12}$$

where N_{active} stands for the number of active slots in a frame (one transmit slot and some receive slots). In the case that each node listens in all slots in the active part of the frame (except its own transmit slot), then we have $N_{active} = N_{listen} + 1$. In the example of one second TDMA frames with 16 active slots, the idle energy consumption of the MyriaNed radio in one frame is $E_{idle} = 52\mu$J.

The radio current in receive mode is greater than that in transmit and idle modes. This is also the case for many other low-power radio chips such as the popular Chipcon CC2420 [6]. The listening time per slot is also longer than the transmission time, as the receiver nodes listen to the whole slot. Thus, listening to the channel consumes more energy than other radio activities. This emphasizes the importance of optimizing listening activities of the node and avoiding unnecessary listening to the wireless channel.

2.4.3 Simulation Framework

We implemented all the proposed protocols and mechanisms in the MiXiM [47] framework on top of the OMNeT++ 4 [5] network simulator. OMNeT++ is an open source discrete event simulator that has a rich modular component-based C++ library for designing frameworks for simulating wired or wireless networks. OMNeT++ is nowadays a widely used simulation platform in networking community. There are several frameworks provided by the community on top of OMNeT++, each well-suited for simulating specific networks. As examples, we mention the INET framework [1] that provides many protocols for wireless networks (e.g., UDP, TCP/IP, Ethernet, 802.11, etc.), INETMANET [2] based on the INET framework for Mobile Ad-Hoc Networks (MANETs), Veins [7] for vehicular networks (VANETs), and MiXiM [47] for wireless mobile sensor networks.

The MiXiM framework provides detailed models of the wireless channel, node mobility, and several networking protocols for wireless mobile networks. It also provides templates and base modules for developing different layers (e.g., physical, MAC, routing, etc.) of network protocols, which further simplify implementing a WSN protocol stack in the OMNeT++ simulator. It makes it suitable for implementing, debugging, and testing our mechanisms.

For each simulated network deployment, we distribute the static nodes in a square area. The nodes are placed around fixed grid points with a 10% variation. This ensures an even distribution of the nodes across the deployment area with a degree of non-uniform placement. It prevents network disconnection by avoiding very sparse or dense deployment in different parts of the area. To evaluate the performance of each mechanism proposed in this thesis, we try several network sizes and densities. We also perform simulations having various numbers of mobile clusters (WBANs). It helps us to observe the behavior of the network with different scales and traffic loads. Each mobile cluster contains several nodes and uses the MoBAN mobility model described in Chapter 3. As the MiXiM framework had no support for group mobility for WBANs, we integrated our developed mobility model for WBANs into the last release of the MiXiM framework.

The wireless sensor nodes in a WSN have some stochastic behavior. Because of that, the result of one simulation run may considerably deviate from the real performance of the network in the long run (the mean value of the underlying random process). To have statistically more reliable results, we repeat every simulation several times. The presented simulation results in this thesis are the average over all simulation runs. We may also investigate the distribution of the results in different simulation runs and check how the results of individual runs deviate from the average results. It is clear that having more runs leads to more reliable performance evaluation. However, because each simulation takes a rather long time to finish, we are not able to afford very many simulation runs.

We use the same mobility pattern for different runs of a simulation setup. Mobility has a great influence on the node connectivity in the network and thus the result of simulation runs with different mobility patterns of the WBANs may substantially deviate from each other. On the other hand, it is not feasible to run very many simulation runs to try a comprehensive set of mobility patterns. Therefore, we use the same mobility pattern when comparing the performance of different mechanisms. The developed WBAN

mobility model (MoBAN) provides an option to generate a mobility pattern and record it, or to use a previously recorded mobility pattern.

The characteristics of the MyriaNed nodes are used as the physical layer settings in simulations. Carrier frequency, transmit power levels, receiver sensitivity, and MAC layer timings are set to be consistent with MyriaNed nodes. A path-loss model [74] is used to model signal attenuation. Using the transmission power of the packet and the calculated value of the signal attenuation, the strength of the signal at the location of different receiver nodes is estimated. It is then compared to the receiver sensitivity to decide if the packet is received by the receiver.

Chapter 3

Mobility Modeling

3.1 Overview

There are a lot of research activities going on for designing proper architectures and protocols for communication in WSNs with some kind of node mobility. The research is being conducted on various aspects of different network layers (e.g., physical layer, MAC, and link layer). A good mobility model is an essential prerequisite for performance evaluation of protocols for such networks and has a big impact on the accuracy of simulations. Mobility models try to mimic the real movement of mobile nodes by characterizing stochastic patterns of node movement. The right mobility model depends on the application scenario.

In a WBAN, sensor nodes are deployed on different positions of a body. Because of the movements of the body, there is high node mobility in the WBAN and the network topology varies frequently over time. The WBAN topology may completely change because of posture changes. The channel quality and connection between nodes strongly depends on the relative position of sensor nodes. It makes the accuracy of the mobility model for WBANs more critical. There is individual mobility of sensor nodes deployed on different positions of the body as well as global movement of the whole body in the surrounding environment. A mobility model should capture both aspects. Furthermore, it should provide sufficient flexibility for researchers to adapt it to their specific application scenarios within the WBAN domain.

This chapter presents MoBAN (Mobility Model for BANs), a comprehensive and configurable mobility model for simulating movement in wireless body area networks. The model is useful for simulating both intra- and extra-WBAN protocols. The former considers the communication between the sensor nodes within a WBAN on a body. Extra-WBAN protocols take care of communications between a WBAN and its environment, with potentially several body area networks as well as an ambient network. The model is configurable, which makes it usable for a large variety of applications. The MoBAN mobility model was published in [60] and this chapter is mainly based on this publication.

There are several open source frameworks well structured for simulating protocols for WSNs and WBANs. Some examples of such frameworks on top of the OMNeT++ [5] discrete-event simulator were mentioned in Chapter 2. There are also several frameworks on top of other network simulators like NS-2/3 [3]. However, there is no freely available WBAN mobility model for simulating WBANs in any of these simulation frameworks. We have implemented our mobility model as an add-on to the mobility framework of the OMNeT++ network simulator so that it can be used for research on WBAN protocols. The MoBAN mobility model is integrated in the MiXiM-2.1 release.

The next section presents a survey on general mobility models for wireless ad hoc and sensor networks. Our mobility model, MoBAN, is presented in Sec. 3.3 and its implementation structure is described in Sec. 3.4. In Sec. 3.5, we point out some cases in which we use MoBAN for simulating our proposed protocols in this thesis. It discusses the need for a mobility model and the usability of MoBAN for performance evaluation of different communication protocols in networks with WBANs.

3.2 Classification of Mobility Models

Several mobility models have been presented in the literature of wireless ad hoc and sensor networks. A detailed survey of early proposed models is presented in [20]. This section firstly classifies mobility models and then briefly reviews some existing models of each class. There are many application-specific mobility models in the literature, but we only review general and commonly used models. Finally, the only existing mobility model for wireless body sensor nodes is reviewed.

In general, we can classify mobility models into two major classes, namely *singular node* and *group* mobility models [83]. In the former class, there is no correlation between the movement of different nodes and it models individual node mobility patterns regardless of the mobility of the other nodes in the network. The latter one takes a group of nodes into consideration that potentially have a particular relationship that introduces correlation between their positions. Social activities of human beings are an example of group mobility for ad-hoc wireless mobile networks.

3.2.1 Singular Node Mobility Models

The Random Walk Mobility Model (RWMM) [109] is a commonly used singular node mobility model in which a node uniformly randomly selects an arbitrary direction and a velocity value from a given range. The node then moves either with a specified fixed time interval or until a given distance is traveled. The node then repeats the random selection and movement process. The movement takes place within a given rectangular space, which is the simulation area. If the node reaches a boundary of the simulation area, it continues along a new path with an angle determined by the direction of hitting the border.

The Random Waypoint Mobility Model (RWPM) [35] is an adapted version of RWMM in which a pause time is inserted between the changes in direction and speed. The pause

time is selected randomly from a given range. In RWPM, a destination is selected randomly from the simulation area. The node then moves toward that position with a randomly chosen speed. All random selections are primarily done using uniform distributions. A specific relation between the pause time and the speed can be applied based on a specific application, to fit the model towards either a more stable network or a network with frequent topology changes.

Both the RWMM and the RWPM suffer from the node concentration problem. Although the destination points are distributed randomly over the simulation area in RWPM, the nodes traverse direct paths toward the selected destination and with a high chance they pass the central part of the area. Because of that, clusters are formed near the center of the area. The Random Direction Mobility Model (RDMM) [84] tries to alleviate the concentration problem and provide a more even distribution of nodes over the entire simulation area. This is done by forcing the nodes to meet a border in each movement step. The node uniformly randomly selects a direction (similar to RWMM) and starts to travel to the border of the area in the selected direction. When the node reaches the simulation boundary, it stops there for a given time. Then the node picks another direction and repeats the process.

All models mentioned so far are memory-less. This means that the completed movement step does not have any impact on the decision about the movement of the next step. In [51], the Random Gauss-Markov Mobility (RGMM) model is proposed in which subsequent movement steps are not independent. Using the Markovian behavior, the speed and direction at each movement step are adapted to determine their value for the next movement step. For adapting the movement parameters, a random Gaussian distribution is used. This model can eliminate sudden stops or sharp direction changes, which happen in the RWMM and RWPM models.

3.2.2 Group Mobility Models

In the wireless networks domain, there are many situations in which there are dependencies between the movement patterns of different nodes. Human mobility patterns mostly show such group behaviors (for instance, because of social activities). Accordingly, several mobility models have been proposed in the literature that each try to model the movement pattern of humans in specific scenarios. The column mobility model, the pursue model, and the nomadic community model, all presented in [86], are some early approaches to model such correlated mobility patterns. Recent activities are mostly based on the concept of higher node density in more popular locations such as the home location. For instance, the Small World In Motion (SWIM) model [55] presents a mobility model based on the fact that humans go more often to locations near their home and to locations in which they can meet many other people. The N-Body [107] mobility model tries to capture the diversity of the distance between different nodes by analyzing real human movement traces. A numerical matrix is produced based on the real movement traces of the target scenario to represent the relationship between each pair of nodes in the network. This information is then used to synthesize new movement traces for simulations.

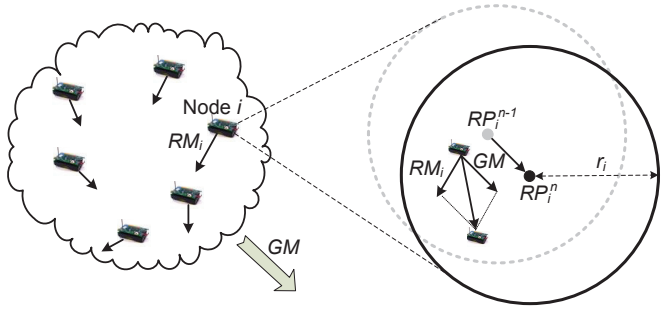


Figure 3.1: Node movement in the RPGM model.

Among all group mobility models, the Reference Point Group Mobility model (RPGM) [34] is a general model that can be tuned to model many scenarios. In fact, many recent proposed models are in some way special cases of RPGM. As we are using this model as a base for our mobility model, it is explained in detail. In the RPGM mobility model, a Logical Center (LC) is defined for the group of nodes, the motion of which defines the entire group movement. Every group has a group motion vector $\overrightarrow{GM}$ that determines the motion of the group's logical center (LC). A Reference Point (RP_i) is assigned to each node i in the group and determines the initial position of the node. The reference point of a node is a fixed point relative to the logical center of the group. Therefore, the reference point of every node within the group moves with the group motion vector $\overrightarrow{GM}$. Fig. 3.1 illustrates the situation for a group. Throughout this chapter, we use the notation $n \in \mathbb{N}$ as the sequence index of the movement steps. In a network simulator (e.g., OMNeT++), it refers to the steps using which the simulator updates the physical position of the nodes within the simulation area. Every node i moves within a predefined area (a circle or sphere with radius r_i) around its reference point with a random motion vector $\overrightarrow{RM}_i$. So there is independent motion of individual nodes in the group while the logical center provides the whole group movement. The motion of every node at each time step is defined by the sum of the group motion and the random motion vectors.

RPGM determines the correlation between nodes in the group by means of a common group motion vector GM. However, it does not prescribe a mobility pattern for moving the LC or for individual movement of nodes within the group ($\overrightarrow{RM}_i$). Therefore, it provides the possibility for the users to specify their desired model for global and individual movements. The appropriate patterns should be designed according to the requirements of the scenario. By selecting a proper group motion behavior, we can model human movement, while setting the random motion vectors ($\overrightarrow{RM}_i$) to define the motion of individual sensors installed on various positions on the body.

The RPGM model has been utilized in [75] to make a model of the mobility of body sensor nodes. In this model, the global movement of the LC (human movement) is done using the Random Gauss-Markov Mobility model. A swarm behavior inspired model

presented in [43] and [44] is utilized for the individual movement of sensor nodes on the body. In this method, if the node is close to its reference point, the model forces it away from that position. On the other hand, if the node is far from its reference point, the swarm inspired method attracts the node towards its reference point. These movements take place within the given range around the reference point of each node.

To the best of our knowledge, the model of [75] is the only mobility model besides MoBAN for wireless body sensor networks described in the literature. However, there are several limitations in this model. First, the model uses certain mobility models for moving the logical center of the group and individual node mobility. It is not clear why these models should be proper choices for a WBAN. Further, it makes the model unadaptable for different applications and sometimes the model is far from real-world mobility patterns. Second, the model does not specifically implement different postures of a human body (fixed reference points). Postures are of great importance in WBANs as the network topology may entirely change due the posture changes. Such dynamics may frequently happen in a WBAN. Further, in a real WBAN, movement parameters of the model such as the speed and target positions of the human depend on the posture. Third, the model as it has been presented only considers position and motion in two dimensions, which is unrealistic for WBANs.

In the following, we present the MoBAN mobility model for evaluating intra- and/or extra-WBAN communication with a focus on completeness, configurability, and availability of the model. It can be configured to model the mobility patterns in various application scenarios of WBANs.

3.3 MoBAN: Mobility Model for WBANs

In this Section, we first explain the general structure of our mobility model (MoBAN) and then present its constituent blocks in more detail.

3.3.1 Model Structure

The RPGM [34] model constitutes the basic platform for modeling mobility in a WBAN in our model. In fact, the RPGM model determines the grouping strategy of MoBAN. We extend the RPGM model by introducing postures, which have individual mobility parameters. The MoBAN itself is constructed by two basic control units, which are the *posture selector* and the *global movement module*. On the one hand, the posture selector process determines the current posture at any instance of time. The individual movement (the random motion vector) of every sensor node is subsequently determined according to the selected posture. On the other hand, the global movement process is responsible for controlling the mobility of sensor nodes on the body as a whole (i.e., moving the logical center of the group). The parameters of this movement are of course not independent from the selected posture and the application scenario.

A mobility pattern starts with the posture selector process. Once the posture is selected for the next mobility phase, information related to the selected posture will be

Algorithm 1: The MoBAN structural process.

```
Data:
τ: posture selection step (τ ∈ ℕ)
π^τ: WBAN posture at step τ
A^τ: Simulation area type at step τ
while True do
    π^τ = SelectPosture(π^{τ−1},A^{τ−1});
    if IsStable(π^τ) then
        T_s=SelectDuration() ;
        wait for T_s seconds ;
    else
        Dest^τ = SelectDestination();
        V^τ = SelectSpeed(π^τ);
        Move the LC toward Dest^τ with velocity V^τ ;
        Wait until Dest^τ is reached ;
    end
end
```

retrieved. If the posture has been defined as a stable posture, such as *lying down*, the posture selector process keeps the control and waits till the selected time duration expires. It then selects another posture based on its strategy. In the case of a mobile posture, such as *walking*, the global movement control module starts moving the whole WBAN taking the parameters related to that specific posture into consideration. The movement behavior, like the destination, speed, and the path to that destination, depends on the specific strategy for the application. Once the WBAN reaches the destination point, the posture selector module gets control to select the next posture and the process is repeated. Algorithm 1 presents this process. The posture selection step is denoted by $\tau \in \mathbb{N}$. According to the model process presented in Algorithm 1, the duration of the posture selection steps is not constant and varies over time because of the randomness in selecting the destination points and the movement speed in mobile postures, and random selection of posture duration for stable postures. Note that the notation n used in this chapter refers to movement steps. One posture selection step may last for an arbitrary number of movement steps. The rest of this section explains the different functions in Algorithm 1.

3.3.2 Posture Specification

Let $C = \{c_1, c_2, \ldots, c_{|C|}\}$ be the set of sensor nodes in the WBAN. Moreover, let $\pi = \{\pi_1, \pi_2, \ldots, \pi_{N_p}\}$ be the set of N_p different postures defined in the mobility model. Each posture is specified by a set of parameter values that are application dependent and should be specified by the user. The specification of each posture includes relative node positions (reference points), the radius of a sphere around the reference point of any node, the velocity of the local movement within the sphere for each node in the WBAN, and the speed range of the whole body movement (global movement). Table 3.1 presents the notations that are used for posture specification parameters in this chapter.

Table 3.1: The basic specification of postures π_j

Notation	Description ($1 \leq i \leq |C|, 1 \leq j \leq N_p$)
$RP_i(\pi_j)$	Relative reference point of node c_i in postures π_j
$V_i(\pi_j)$	Velocity of node c_i
$r_i(\pi_j)$	Radius of movement sphere of node c_i
$V_{min}(\pi_j)$	Minimum velocity of the WBAN
$V_{max}(\pi_j)$	Maximum velocity of the WBAN

A posture π_j can be set as mobile or stable by specifying the value of the maximum possible speed $V_{max}(\pi_j)$ of its global movement (zero for stable postures). This is actually specified according to the application scenario. In general, typical mobile postures are *walking* and *running*. But in specific applications such as a hospital situation, *sitting* or even *lying down* can be thought of as mobile postures when some sort of carrier like a wheelchair or mobile bed is used.

3.3.3 Local Mobility

Individual movement of any sensor node within the WBAN depends on the selected posture. Based on the RPGM model, every node moves within a given sphere around its reference point. The local movement parameters, which are speed $V_i(\pi_j)$ and radius $r_i(\pi_j)$ of the movement sphere around the reference point, are obtained from the specifications of the selected posture π_j shown in Table 3.1. The values are given based on the expected movement of the node deployed on that specific position in the current posture. A node on an arm can have a very high mobility in the *running* posture whereas it has a very low mobility in the *lying down* posture. A node deployed on the chest is fixed and has no local mobility (zero velocity) in alll postures.

We use the random waypoint mobility model (RWPM) [35] in 3D space to determine the random motion vector $\overrightarrow{RM}_i$. In each step, a point within the sphere space is selected uniformly randomly as the destination. The node then moves toward the destination with the given velocity value $V_i(\pi_j)$.

3.3.4 Posture Selection Strategy

Posture selection is a very important part of the model as the posture influences the network topology and node connectivity. The selected posture also determines the local movement of nodes and the global mobility of the whole WBAN. Therefore, it affects both the connection between the nodes within the WBAN and the connection with the external network like other WBANs or the surrounding ambient network. Experimental measurement results in [72] show how the quality of the wireless links between different sensor nodes deployed on a human body vary over different postures. As the communication protocols are expected to support such changes in the network topology and link quality, the mobility model should model posture changes to evaluate the protocols.

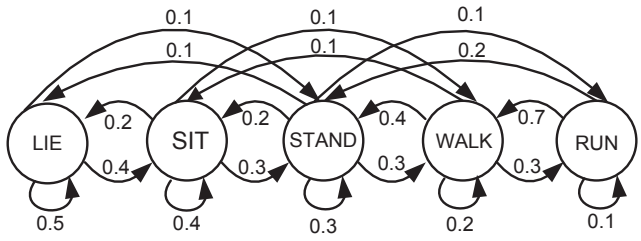


Figure 3.2: A (typical) Markov chain for posture pattern selection.

A posture pattern is a sequence of selected postures in a certain period of time (for example, the whole simulation time). Statistically, in reality, some posture patterns take place with a higher probability than other patterns. For example, changing the posture from *lying down* to *running* is very rare whereas changing to the *sitting* posture is more likely. We use a one-level discrete-time Markov chain to model posture sequences. There is also the possibility to use multiple Markov chains to differentiate between area types (see below for details). The Markov model is described by a transition probability matrix P in which $0 \leq P_{ij} \leq 1$ stands for the probability of the transition from posture π_j to π_i. In a Markov model, transitions originating from a specific state (posture) should add up to one.

$$\sum_{i=1}^{N_p} P_{ij} = 1 \qquad for\ all \quad 1 \leq j \leq N_p \tag{3.1}$$

The transition matrix (P) can be obtained from real human posture traces. Fig. 3.2 shows a typical Markov chain including five different postures, namely *running*, *walking*, *standing*, *sitting*, and *lying down*. The edges are labeled with the transition probabilities.

An important parameter is the time duration of each posture. In the case of a mobile posture, the time duration is indeed built-in as the next posture selection is done upon reaching the selected destination. If the current posture is a stable posture, the posture selector process waits for a certain duration and then selects the next posture according to the Markov model. We ask the user to specify a desired distribution for the time duration of each posture according to the application scenario. This can simply be a constant time duration or a uniform distribution within a given range, or a more precise distribution closer to real-life posture durations. The function *SelectDurations*() in Algorithm 1 uses the given distribution to select the posture time duration upon the selection of a stable posture.

3.3.5 Location and Posture Pattern Coherency

Posture selection based on a Markov model is the basic abstraction in our model. Given the Markov model, we decide about the next posture considering the current posture and the transition probability matrix. However, for many applications, the likelihood of posture patterns may depend on the location in the area of simulation. To take this correlation between the WBAN position and the posture pattern into account, we provide the user of the model with different abstraction levels for modeling. Let $A = \{A_1, A_2, \ldots, A_{N_d}\}$ be a partitioning of the simulation area into N_d different area types. In any area type, we may have different posture patterns. As an example, different rooms in a building can be thought of as having statistically different posture patterns. The posture pattern in a bedroom is surely different from the pattern in the living room or the kitchen.

For the most precise level of specification, a user can define a dedicated Markov model for any location of the area. In this case, N_d different transition probability matrices should be given.

At a coarser level of specification, we just ask a user to specify a Markov chain with a transition matrix (P) for posture changes plus the steady-state probability vector (Π^{A_k}) that determines the distribution of postures in each area type. Thus $0 \leq \Pi_i^{A_k} \leq 1$ stands for the steady-state probability of posture π_i in area type A_k. According to the properties of the steady-state vectors, the sum of all elements should be one, as shown in Eqn. 3.2.

$$\sum_{i=1}^{N_p} \Pi_i^{A_k} = 1 \tag{3.2}$$

We can then automatically find a Markov matrix with the desired steady-state distribution. Consequently, the transition matrix P^{A_k} for any area type A_k is extracted by adapting the initial transition matrix P to satisfy the desired steady-state vector Π^{A_k}.

So, on the one hand, the goal is to find a transition probability matrix (Markov chain P^{A_k}) with the steady-state probability vector Π^{A_k}. Based on the definition of steady-state probability vector, the target transition matrix should satisfy Eqn. 3.3.

$$P^{A_k} \cdot \Pi^{A_k} = \Pi^{A_k} \tag{3.3}$$

On the other hand, the resulting transition matrix should be as much as possible similar to the initial transition matrix P. We translate this to minimizing $\| P^{A_k} - P \|_F$ where $\| \cdot \|_F$ stands for the Frobenius norm (which is a standard norm for matrices). In other words, if we let $D = P^{A_k} - P$, the aim is to find a minimal $\| D \|_F$ such that

$$D \cdot \Pi^{A_k} = P^{A_k} \cdot \Pi^{A_k} - P \cdot \Pi^{A_k} = \Pi^{A_k} - P \cdot \Pi^{A_k} \tag{3.4}$$

If ${\Pi^{A_k}}^{\dagger}$ is the pseudo inverse of Π^{A_k}, the minimum (least) norm solution for D (D_{ls}) is as follows.

$$D_{ls} = (\Pi^{A_k} - P \cdot \Pi^{A_k}) \cdot {\Pi^{A_k}}^{\dagger} \tag{3.5}$$

$\Pi^{\mathrm{A_k}}$ is a vector with dimension $N_p \times 1$, so its pseudo inverse can be calculated as follows.

$$\Pi^{\mathrm{A_k}\,\dagger} = \Pi^{\mathrm{A_k}\,T}\left(\Pi^{\mathrm{A_k}} \cdot \Pi^{\mathrm{A_k}\,T}\right)^{-1} = \frac{\Pi^{\mathrm{A_k}\,T}}{\parallel \Pi^{\mathrm{A_k}} \parallel^2} = \frac{\Pi^{\mathrm{A_k}\,T}}{\sum_{i=1}^{N_p} (\Pi_{\mathrm{i}}^{\mathrm{A_k}})^2} \tag{3.6}$$

As we assumed that $\mathrm{D} = \mathrm{P}^{\mathrm{A_k}} - \mathrm{P}$, then the solution for the new transition matrix is calculated by Eqn. 3.7.

$$\mathrm{P}^{\mathrm{A_k}} = \mathrm{P} + \mathrm{D} = \mathrm{P} + \frac{1}{\parallel \Pi^{\mathrm{A_k}} \parallel^2}(\mathrm{I_{N_p}} - \mathrm{P}) \cdot \Pi^{\mathrm{A_k}} \cdot \Pi^{\mathrm{A_k}\,\mathrm{T}} \tag{3.7}$$

where $\mathrm{I_N}$ stands for the identity matrix of size N.

As the resulting transition matrix $\mathrm{P}^{\mathrm{A_k}}$ should be Markovian, every column of the extracted matrix $\mathrm{P}^{\mathrm{A_k}}$ should add up to one. Calculations in Eqn. 3.8 verify that this requirement is always satisfied by Eqn. 3.7.

$$\begin{aligned}
\sum_{i=1}^{N_p} P_{ij}^{A_k} &= \sum_{i=1}^{N_p}\left(P + \frac{1}{\parallel \Pi^{\mathrm{A_k}} \parallel^2}(I_{N_p} - P) \cdot \Pi^{A_k} \cdot \Pi^{A_k\,T}\right)_{ij} \\
&= \sum_{i=1}^{N_p}\left(P_{ij} + \frac{\Pi_i^{A_k}\Pi_j^{A_k}}{\parallel \Pi^{\mathrm{A_k}} \parallel^2} - \frac{1}{\parallel \Pi^{\mathrm{A_k}} \parallel^2}(\mathrm{P} \cdot \Pi^{A_k})_i \Pi_j^{A_k}\right) \\
&= \underbrace{\sum_{i=1}^{N_p} P_{ij}}_{1} + \frac{\Pi_j^{A_k}}{\parallel \Pi^{\mathrm{A_k}} \parallel^2}\underbrace{\sum_{i=1}^{N_p}\Pi_i^{A_k}}_{1} - \frac{\Pi_j^{A_k}}{\parallel \Pi^{\mathrm{A_k}} \parallel^2}\sum_{i=1}^{N_p}\left(\sum_{j=1}^{N_p} P_{ij}\Pi_j^{A_k}\right) \\
&= 1 + \frac{\Pi_j^{A_k}}{\parallel \Pi^{\mathrm{A_k}} \parallel^2} - \frac{\Pi_j^{A_k}}{\parallel \Pi^{\mathrm{A_k}} \parallel^2}\sum_{j=1}^{N_p}\left(\Pi_j^{A_k}\sum_{i=1}^{N_p} P_{ij}\right) \\
&= 1 + \frac{\Pi_j^{A_k}}{\parallel \Pi^{\mathrm{A_k}} \parallel^2}(1 - \underbrace{\sum_{j=1}^{N_p}\Pi_j^{A_k}}_{1}) \\
&= 1 \qquad \blacksquare
\end{aligned} \tag{3.8}$$

The properties of the transition probability matrices (Eqn. 3.1) and steady-state probability vectors (Eqn. 3.2 and Eqn. 3.3) are used in the above formulation.

Besides the proven requirement about the sum of elements in every column, all entries of the transition matrix should be a probability value ($0 \leq P_{ij}^{A_k} \leq 1$). However, Eqn. 3.7 does not necessarily satisfy this requirement. We take that into account for computing the matrix. To do so, it may be necessary to modify the resulting matrix and repeat Eqn. 3.7 in a recursive manner till we obtain a proper transition matrix satisfying the desired steady-state probabilities ($\Pi^{\mathrm{A_k}}$).

As an example, consider the initial Markov model of Fig. 3.2 with the transition matrix P given in Eqn. 3.9.

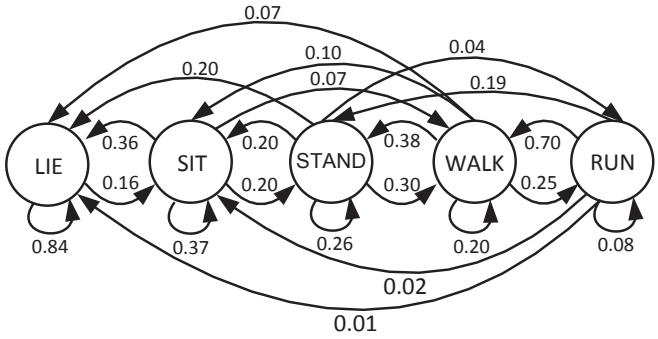


Figure 3.3: The updated Markov chain of the initial chain of Fig. 3.2 for posture pattern selection in a certain area type A_0 with the given steady-state vector of posture distribution Π^{A_0}.

$$\mathbf{P} = \begin{pmatrix} 0.5 & 0.2 & 0.1 & 0 & 0 \\ 0.4 & 0.4 & 0.2 & 0.1 & 0 \\ 0.1 & 0.3 & 0.3 & 0.4 & 0.2 \\ 0 & 0.1 & 0.3 & 0.2 & 0.7 \\ 0 & 0 & 0.1 & 0.3 & 0.1 \end{pmatrix} \tag{3.9}$$

Now, suppose that we specify a steady-state posture distribution Π^{A_0} for the area type A_0 as in Eqn. 3.10.

$$\mathbf{\Pi^{A_0}} = \begin{pmatrix} 0.6 \\ 0.2 \\ 0.1 \\ 0.075 \\ 0.025 \end{pmatrix} \tag{3.10}$$

This steady-state vector can be thought of as a distribution for a bed room, for example. It says that if the person is in the bedroom (area type A_0), in 60% of the time she/he lies down, while the probability of running is very low.

The new Markov model P^{A_0} for this area type is calculated using Eqn. 3.7 as follows.

$$\mathbf{P^{A_0}} = \begin{pmatrix} 0.84 & 0.36 & 0.20 & 0.07 & 0.01 \\ 0.16 & 0.37 & 0.20 & 0.10 & 0.02 \\ 0 & 0.20 & 0.26 & 0.38 & 0.19 \\ 0 & 0.07 & 0.30 & 0.20 & 0.70 \\ 0 & 0 & 0.04 & 0.25 & 0.08 \end{pmatrix} \tag{3.11}$$

Fig. 3.3 shows the resulting Markov chain for the area type A_0. This model satisfies the given steady-state postures distribution Π^{A_0} (Eqn. 3.10) and it still takes the more likely transitions and posture patterns into account based on initial Markov chain P.

This way the model can be configured to be very specific for a particular scenario by providing one Markov chain for each area type of the simulation area, or it can be more general through utilizing a less precise model specification, which will be more convenient for the user.

3.3.6 Global Movement

When a mobile posture is selected, a target position should be chosen to start movement of the whole WBAN (moving the *LC* of the group). A uniform random strategy can be applied as the basic level if the probability of being selected as the target position for all area types of the simulation area is the same in the application. There is also the possibility to have a specific non-uniform WBAN position distribution for different area types of the simulation area. For instance, we may specify the probability of the area type A_i being selected as the target position for the movement of the WBAN.

Different WBANs in the network move independently from each other. However, in many applications, modeling social activities (e.g., meeting in a room) is of great importance as it can change many things in the network, like interference, wave propagations, and the connectivity between WBANs and the ambient WSN infrastructure. According to the application scenario, a community-based mobility model existing in the literature of mobile ad-hoc wireless networks (e.g. models presented in [50, 55, 86, 107]) can be chosen to be integrated with MoBAN to include social activities as well.

3.3.7 Temporal Correlation

The spatial correlation has been taken into account in MoBAN through the possibility of specifying location-dependent distributions for the posture pattern, and WBAN target position selection, as described. However, in many applications in reality, time is also important. For instance, the area types that a human (WBAN) may visit during day time and night time are different. As a solution, one can independently conduct several simulation runs with different parameters and distributions to check the performance of the network in different time frames. Nevertheless, integrating such a facility in the mobility model is worthwhile and makes the model more convenient to use.

Temporal correlations can be accordingly integrated into the model by performing the space and time partitioning with the same mechanism that was already explained for partitioning area. It means both the target position and the posture pattern selection processes can be done taking the time into consideration as well. To do so, the simulation area is partitioned into N_d different area types, as explained. Each area type then is partitioned into N_t separate time frames. By this, we have $N_d \times N_t$ space-time partitions in which $A_{i,j}$ is set to be the i^{th} spatial area type in the j^{th} time frame. Now, different distributions can be specified for different space-time partitions based on the application scenario and the required level of precision.

3.3.8 WBAN Radio Model Parameters

The human body has a severe influence on radio wave propagation and it affects the quality of the wireless links between different nodes in a WBAN. Thus it is very important to take the body effect on propagation loss and link quality in various postures into consideration. As the effect of the body depends on the relative position of nodes and the body situation, it is very useful to include that in the WBAN mobility model as the mobility model is responsible for the position of nodes at any instance in time.

Based on the radio propagation model that is used for network simulation, we can decide about the channel parameters in the mobility model according to the current situation of the body. Provided that a path-loss model is used, for instance, we specify for every pair of nodes in the definition of each posture, the mean (μ_α) and deviation (σ_α) of the path loss coefficient α. The mean value is specified (in the range from 3 to 7, see [106]) according to the ratio of the distance between the pair of nodes in which radio waves should be propagated around or through the body. The deviation is set based on the relative mobility of the pair of nodes in the specific posture. The proper values for the mean and deviation parameters can be extracted from the result of a real experiment. An experimental WBAN radio propagation study for several subjects in various postures is presented in [88]. It provides a path-loss model characterization for each subject and posture by analyzing the experimental data. This characterization can be used to set more realistic values for modeling radio channel in the target application. During the simulation run, a value is selected for α according to the normal distribution $\mathcal{N}(\mu_\alpha, \sigma_\alpha)$ at the start time of every selected posture.

3.4 MoBAN Implementation

We implemented the MoBAN mobility model as an add-on to the mobility framework of the OMNeT++ discrete event simulator, to be used for our own research as well as to make it available to the scientific community for other research on WBAN protocols. The implementations of the WBAN mobility model as well as the pure RPGM group mobility model are available through the web site *http://www.es.ele.tue.nl/nes/*. It is also integrated in MiXiM-2.1 release. Any number of WBANs can be instantiated and different WBANs can include different numbers of sensor nodes. The ambient network (non-WBAN nodes) can still be involved and can have their own mobility using existing singular node mobility models existing in the mobility framework of OMNeT++. By this, a complete combined network consisting of several WBANs and an ambient sensor network can be simulated.

3.4.1 MoBAN Implementation Modules

Fig. 3.4 shows the block diagram of the implementation of the mobility model for a single WBAN. To have a WBAN in the network, one mobility coordinator module is instantiated in the top level simulation setup as well as one mobility module inside each node within the WBAN. In the case of multiple WBANs, multiple coordinators should be

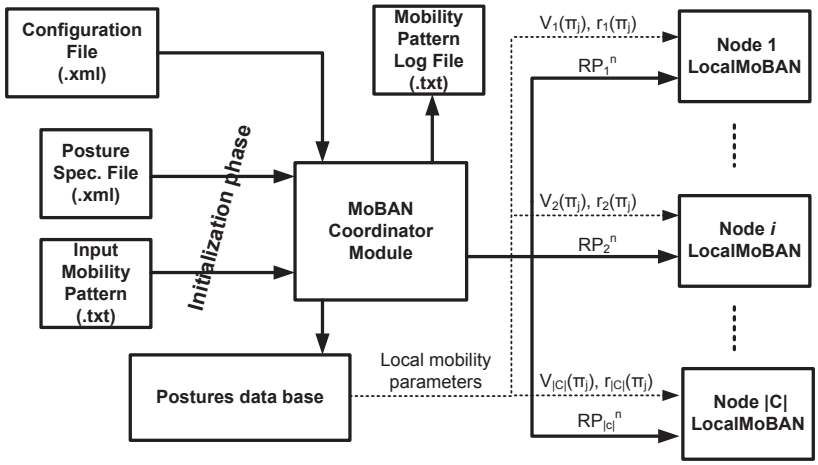


Figure 3.4: Block diagram of the OMNeT++ implementation of the MoBAN mobility model. π_j is the current posture.

instantiated and the mobility module in each sensor node specifies the enclosing WBAN coordinator module. More information about the C++ implementation of these modules is available in the documents that come along with the MiXiM implementation package of MoBAN.

The mobility coordinator is the main module that provides the group mobility and correlation between nodes in a WBAN. In the initialization phase, it reads several user defined files. First of all, it reads the postures specification file and makes a database of that information to be used during simulation by the coordinator module itself or by other nodes. Any node can access the posture database during simulation to retrieve information about its local mobility parameters (current posture, $V_i(\pi_j)$, and $r_i(\pi_j)$). A configuration file as well as all required distributions for specifying different non-uniform distributions is also read in the initialization phase. Note that all WBANs may use a single posture or configuration file if they are in the same situation. However, different input files can also be specified to have variety between different WBANs in a simulation run.

During the simulation, the mobility coordinator decides about the posture and the global movement of the whole WBAN (position of the LC) by implementing the *posture selector* and *global movement* processes of the model based on Algorithm 1. The LC's position of the WBAN is an absolute position within the (three-dimensional) simulation area and is determined according to the mobility model that is being used for movement of the whole WBAN. The coordinator module also knows the reference point (RP) of each node in the current posture as well, as a relative position to the LC. What a node c_i within the WBAN needs is actually its absolute physical location of the reference

point (RP_i^n) as well as the current posture (π_j). RP_i^n is calculated by adding the current position of the logical center (LC^n) by the relative position of the reference point $RP_i(\pi_j)$ of the node c_i in posture π_j ($RP_i^n = LC^n + RP_i(\pi_j)$). The mobility coordinator sends the new RP_i^n to every node belonging to the WBAN.

The local mobility module of a sensor node c_i receives the RP_i^n and the current posture π_j of the WBAN from its coordinator module. Subsequently, it retrieves the parameters of the local movement ($V_i(\pi_j)$, $r_i(\pi_j)$) from the posture database. It then uses RWPM with the given local movement parameters of the current posture to determine the physical position of the node. Note that the individual node mobility module has nothing to do with a posture change as its reference point is properly updated upon a posture change by the coordinator module. It only retrieves the local movement parameters of the new posture.

3.4.2 Recording and Reusing Mobility Patterns

Sometimes we simulate a protocol to compare its performance with alternative protocols. In such a situation, to have fairer comparison, it is worthwhile to use the exact same mobility pattern for simulating different protocols. In our implementation of the WBAN mobility model, we have the feature of logging the mobility pattern of a WBAN. The coordinator module logs the selected destinations, velocity values and posture patterns in a file if the logging function is requested. The coordinator module can be set to read a previously logged mobility pattern of an earlier simulation run by specifying the name of the input mobility pattern file. To simulate different protocols for a given network, we may run the first simulation and log the mobility pattern. Then, for the rest, we use the logged pattern.

3.5 MoBAN Utilization

We used MoBAN to simulate the communication protocols that are proposed in this thesis. These protocols include both intra-WBAN communication protocols and protocols supporting the combination of an ambient static network and several WBANs. MoBAN with specific configurations is usable for simulating both kinds of networks. In this section, we briefly present some simulation setups and results to illustrate the potential uses of MoBAN. The setups are actually used in later chapters for detailed analysis of the performance of the proposed communication mechanisms.

For simulating an intra-WBAN communication protocol, only one instance of the MOBAN coordinator module is sufficient in the simulation setup. We used twelve sensor nodes including a gateway node installed on different positions of a body. Accordingly, the specifications of five different postures are defined and the initial Markov model of Fig. 3.2 is used to generate the posture pattern. We used the mobility pattern recording and reuse capability of the MoBAN implementation to have the same mobility patterns for conducting simulations using different runs. Fig. 3.5 shows snapshots of this simulation setup in the OMNeT++ GUI in different postures. The lines in the figure show potential wireless links between sensor nodes.

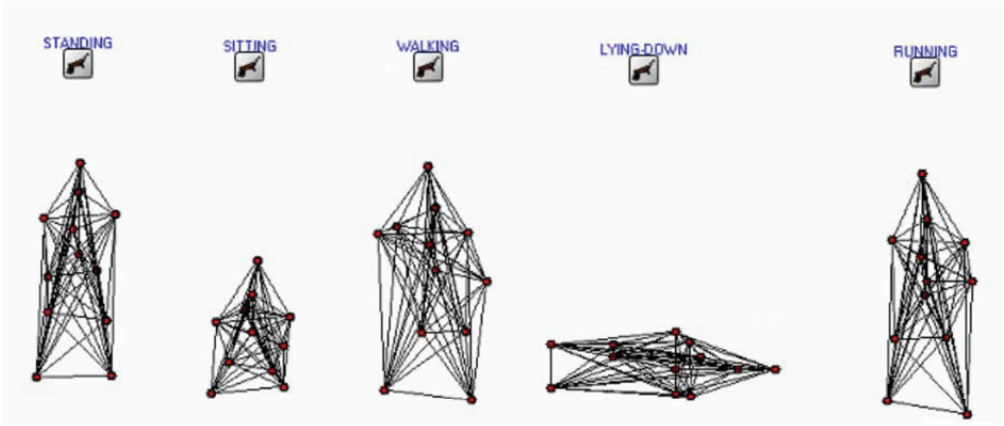


Figure 3.5: Snapshots of OMNeT++ simulation for intra-WBAN communication using MoBAN in different postures (front view).

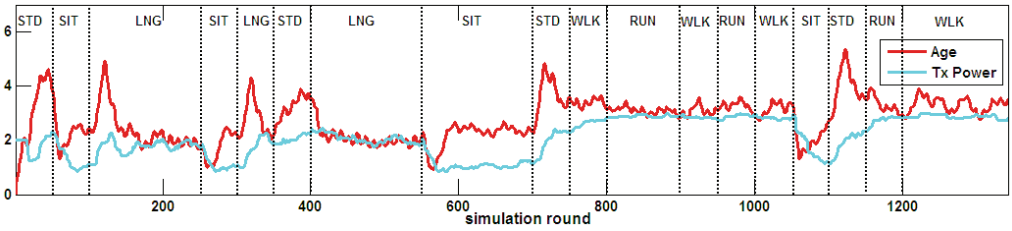


Figure 3.6: Average *age* of data items at the gateway and average transmit power of a node in simulation using MoBAN

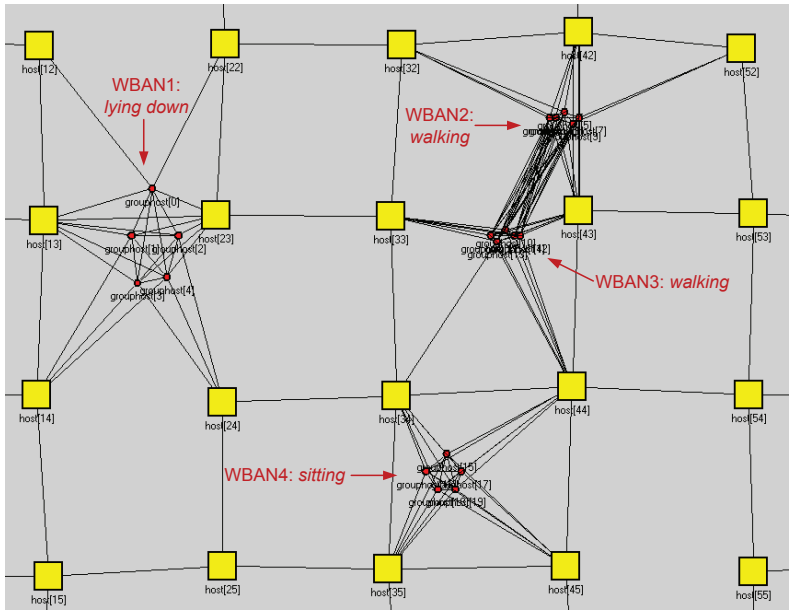


Figure 3.7: A snapshot of the simulation for extra-WBAN communication using MoBAN; view from above in OMNeT++ simulation GUI.

All sensor nodes need to send their data items to the gateway node on the body directly or through a multi-hop data forwarding. A transmit power adaptation mechanism is proposed to optimize the transmit power consumption of sensor nodes while realizing the proper node connectivity to meet the quality-of-service requirements. Details about the power adaptation protocol are available in [57]. Fig. 3.6 shows a part of the simulation result in which the average age of data items at the gateway node is shown. The sensitivity of the network performance (the average age) to the posture changes as well as the reaction of the protocol via changing the transmit power of nodes are visible in the figure. It is best visible in changing from closed postures (e.g., *sitting*) to open postures (e.g., *standing*) and vice versa. Abrupt changes in the value of the average age reveal the impact of posture changes on the connectivity of the WBAN nodes. Transmit power adaptation tries to keep the age level within a desired range. The transmit power is selected among four levels, denoted 0, 1, 2, and 3. The graph shows the average over the 32 runs.

For simulating the combined static and WBANs network, we made a simulation setup of 100 ambient (static) sensor nodes placed around fixed grid points as well as four WBANs each including five sensor nodes. WBANs move independently within the simulation area using the MoBAN implementation. Therefore, four instances of the MoBAN

coordinator module are used. Fig. 3.7 depicts what the simulation looks like in OMNET++ at a particular point in time. Four WBANs as well as their current postures are shown. *WBAN2* and *WBAN3* are in the potential interference range of each other. So they may interfere or hear each other. This affects the performance of the system and is important in the investigation of the behavior of the proposed protocol in such situations. Simulation results presented in Chapter 4 show that the collision ratio grows with increasing the number of WBANs in the simulation area. The collision ratio definitely depends on the time durations that different WBANs are close together. This means that modeling the WBAN movement according to the target application is of great importance for decisions about the applicability of the protocol. The current available implementation of MoBAN uses random waypoint mobility model for global movements of WBANs. Other models may also be integrated with the current implementation.

3.6 Summary

This chapter presented MoBAN, a Mobility model for wireless Body Area Networks. The model has been specifically designed so that it can be configured for being used for performance evaluation of a broad range of application scenarios including WBANs. Both global movement of the WBAN and the individual node mobility within the WBAN have been taken into consideration. The model can be used in simulating both intra- and extra-WBAN protocols. Two use cases were described that shows the usability of the model in simulating different kinds of networks including WBAN. The implementation of the model as an add-on to the mobility framework of the OMNeT++ event simulator makes the model available to the scientific community to be used for research on WBAN communication protocols and applications. The implementation of MoBAN as well as the RPGM group mobility model on top of the OMNeT++ simulator can be obtained through *http://www.es.ele.tue.nl/nes/*. It is also integrated in the MiXiM-2.1 release.

Chapter 4

Medium Access Control for Mobile Clusters

4.1 Overview

Wireless sensor nodes in a WSN use a shared wireless channel medium to communicate with each other. When two nodes are in each other's interference range, they should not transmit at the same time to prevent collisions. However, if a collision happens, there should be a mechanism to deal with the situation. The Medium Access Control (MAC) layer is responsible to manage the access of different nodes to the shared medium and resolve the collisions. Traditional MAC protocol design aims to maximize throughput; MAC protocol design for WSNs has minimizing the energy consumption as a main objective. To this aim, radio activities such as listening to the wireless channel, in particular, should be conservatively optimized. However, besides optimizing energy consumption, reliability in communication, latency, and robustness are still requirements of the WSN applications, which need to be provided by the MAC layer.

Thanks to the wireless communication capability, the sensor nodes in WSNs have the opportunity to be mobile. The ability of wireless nodes to move provides additional interesting applications for WSNs. Moreover, node mobility is a key feature of using WSNs in many sensory application domains, such as healthcare. The underlying networking protocol stack, including MAC layer, should sufficiently support expected node mobility in the network. In schedule-based contention-free protocols in particular, accessing the shared medium is scheduled taking the neighborhood into account. Time slots that are dedicated to the nodes should be unique in their neighborhood. This provides an efficient use of the channel bandwidth by avoiding collisions. Because of this, these mechanisms are promising for communication in WSNs. However, with node movement, the neighborhood may change, so the slots need to be dynamically rescheduled. This process is time and energy consuming. Rescheduling mechanisms are well-suited for network startup

and for joining nodes. Static networks with very limited mobility (node relocation) can also be supported. However, applying such processes is really challenging for networks with high node mobility.

The mobility is even more challenging if several nodes in the network move together (cluster mobility). In this case, several nodes together may move to a new neighborhood and cause more overlapping slots that need to be resolved by the slot rescheduling scheme. There are many applications in which a static sensor network is in place together with several groups of mobile nodes. Our intended health application is an example. To achieve a good performance, providing dedicated support for these highly mobile WBAN clusters is of great importance.

Nodes in WSNs have generally very limited resources. Optimizing energy consumption is critical for such networks to provide a reasonable life time for the nodes. The radio activities usually consume most energy of a sensor node. In particular, listening or scanning the medium for communication constitutes a substantial part of the overall energy consumption [91]. Thus minimizing these activities is of great importance to have a low-power WSN. Low duty-cycling is an effective approach to reduce energy consumption by keeping the radio transceivers off and only turning them on periodically in specific time durations to perform radio activities. However, even in the short active durations, unnecessary radio activity should be avoided, because it wastes the energy resources. *Idle listening* to the wireless channel, in particular, is a source of major energy wastage and MAC protocols try to minimize it [24, 67, 69]. Node mobility also plays a role here. When a node expects packets from a specific neighbor, it listens to the channel in the dedicated time slot. However, if the sender has moved to another neighborhood, the listening node does not receive anything. Thus mechanisms to avoid *idle listening due to node movement* will be of great benefit.

This chapter introduces MCMAC (Mobile Cluster MAC), a mechanism to support cluster mobility in networks with a static backbone using a TDMA-based MAC protocol. Here a flat architectures (discussed in Sec. 1.4) is considered for communication of WBAN nodes with the static network. MCMAC is built on the base of a TDMA-based MAC, because of the promising characteristics of the TDMA mechanism such as efficient collision-free access to the wireless channel. MCMAC exploits a contention-based medium access scheme to be used by mobile cluster nodes within the existing low-duty-cycle TDMA mechanism. The protocol is designed to keep the duty cycle of all nodes as short as possible, and to be integrated with our listening scheduling mechanism, which is proposed to reduce idle listening to mobile clusters. Carrier-Sense Multiple-Access (CSMA) and slotted ALOHA are used as two contention-based medium access schemes for mobile cluster nodes. The behavior of both contention-based paradigms is analyzed and guidelines for choosing a contention-resolution mechanism and protocol parameter settings are provided based on the hardware setup, mobile cluster density, and application requirements. The main objective of the MCMAC protocol is to *efficiently deliver the data generated by the mobile nodes to the static network and optimize energy consumption of nodes for performing this task.* A preliminary version of the MCMAC protocol was published in [59].

There are several configuration parameters in the proposed mechanism. We perform an empirical exploration to study the trade-offs due to different parameters of the protocol. This is then used to select near-optimal configurations of the protocol. The protocol is evaluated using real-world experiments as well as computer simulations. The protocol is implemented on real wireless nodes and several large-scale experiments have been performed (explained in Chapter 2) to validate the protocol and investigate its behavior in real-world networks. A static network of 60 nodes was deployed on two floors of an office building. Four volunteers were participating, each carrying four mobile nodes, while performing their daily mobility behavior.

The next section investigates different MAC mechanisms and the way they deal with clustering and mobility. Sec. 4.3 describes the proposed protocol for supporting cluster mobility. The listening scheduling mechanism is explained in Sec. 4.4. The experimental setup and evaluation results are given in Sec. 4.5. Sec. 4.6 discusses an extension of the protocol for crowded networks.

4.2 A Review on MAC Paradigms and Mobility Support

In this section, we review prominent MAC paradigms in WSNs, showing their respective advantages and disadvantages. In particular, their methods of dealing with clustering and mobility are studied.

4.2.1 MAC Paradigms for WSNs

There are several MAC protocols designed for WSNs, many of them targeting specific applications. The prominent MAC protocols for WSNs are explored in [48], in which their respective performance is analyzed. In general, MAC protocols for sharing the wireless channel in WSNs exploit one of the two major paradigms: *contention-based* or *schedule-based*. Carrier Sense Multiple Access (CSMA) and ALOHA are the base mechanisms for designing many contention-based protocols in the literature of wireless ad hoc and sensor networks. Wise-MAC [24] (based on ALOHA) and B-MAC [69] (based on CSMA) are known *asynchronous* protocols that rely on low power listening (LPL) to reduce idle listening. In the LPL mechanism, nodes periodically wake-up and probe the wireless channel. They then return to the sleep mode if they do not detect a signal. Therefore, nodes do not consume energy for continuous listening to the channel for possible transmissions by the sender nodes. However, a transmitter node needs to transmit a long preamble before transmitting a data packet so that the receivers catch the preamble in one of their periodic channel probes. This imposes an overhead on the sender node. X-MAC [17] tries to alleviate the overhead by a shorter preamble using an acknowledge from the waked-up receiver. The sender node stops transmitting preamble when it receives an acknowledge from the receiver. It also reduces the overhearing problem by embedding the target address in the preamble. Thus the receiver nodes stop listening if they are not the target of the packet. S-MAC [104] and T-MAC [22], try to reduce idle listening by use of time frames with active and sleep parts. All communications occur in the active part, which leads to reduction of energy wastage due to idle listening. To

further reduce energy consumption, SCP-MAC [105] shortens the length of the active part of the frames by synchronizing the channel probing (LPL) times of all neighboring nodes. Using scheduled channel probing, very short preamble is required for senders to wake up the receivers in the range.

Contention-based protocols are considered well-suited for WSNs due to their simplicity, flexibility, and no need for much network infrastructure support. No assumption is made on network topology and node neighborhoods. Joining and leaving the network can be done without extra operations. So node mobility natively does not need specific support in this category of MAC protocols. However, these protocols suffer from inefficient use of the bandwidth due to collisions. Therefore, the mechanism is not well suited for dense networks. Although CSMA uses carrier sensing to avoid collisions, they can still happen due to the *hidden terminal* problem. This happens when two nodes are not in the interference range of each other, but their interference ranges overlap. Thus they cannot detect each others transmission by carrier sensing, but they collide at the location of a third node. Some protocols such as T-MAC [22] try to solve this issue by using a Request-To-Send (RTS) and Clear-To-Send (CTS) mechanism. The sender nodes check the channel clearance at the place of the receiver by asking a clear channel confirmation from the receiver. However, such solution is power costly for the rather short data packets in many WSN applications. The switching delay from carrier sensing to transmission mode is another native source of collisions in CSMA-based protocols.

Nodes using a schedule-based protocol, use TDMA to share the medium without collisions. These protocols are more efficient in terms of bandwidth usage. However, there are several issues that complicate these protocols. Most prominently, synchronization of time frames and efficient slot scheduling are important requirements to provide efficient communication. PEDAMACS [26] provides a centralized synchronization and slot scheduling mechanism using a high-powered access point. TRAMA [73], DRAND [79], and LMAC [97], are examples of efforts for efficient distributed slot scheduling in TDMA frames. The LMAC protocol, for instance, provides a distributed mechanism for occupying unique slots in a two-hop neighborhood. Every node propagates a bit-set detailing the occupied slots by itself and its one-hop neighbors. A node can detect the free slots in its two-hop neighborhood by OR-ing the received occupancy bit-sets. The node then selects a random slot among the unoccupied slots. The biggest challenge for such scheduling methods is network dynamics. In particular, node mobility can destroy the schedules and causes collisions. In this sense, TDMA-based protocols are said to be efficient protocols for static networks. Efficient and reliable time synchronization and slot scheduling techniques for TDMA-based mechanisms are still ongoing research topics.

There are also some protocols in the literature that combine the two paradigms to collect their respective strengths. Z-MAC [78], for instance, uses such a hybrid approach. The nodes in this protocol are synchronized. Under a low contention, nodes use CSMA in the TDMA slots and under high contention they try to use their own scheduled TDMA slots. Although Z-MAC is known as the first attempt of using a hybrid TDMA/CSMA approach for WSNs, the concept was previously exploited in [25] for a one-hop wireless LAN environment. The idea is again to switch between TDMA and CSMA according to the contention level observed in the network.

4.2.2 Mobility Support in TDMA-based MAC Protocols

Several *rescheduling* methods have been presented for TDMA protocols with the main objective of supporting joining and leaving nodes. A movement may be treated as leaving one neighborhood and joining another one. In LMAC [97], for instance, a node may cause interference when it moves until the collisions are detected and slots are rescheduled in the neighborhood. Of course, a proper interference detection mechanism should be in place. In [38], slot scheduling is done by cluster heads. Upon a node movement, the moving node may join another cluster and so it will be assigned a new slot by the head of the new cluster.

Such reactive rescheduling methods are effective for joining nodes and also for node relocation (very limited infrequent mobility). For instance, sensor nodes that are mounted on furniture can move and use this method to reschedule their transmit slot. Movement detection and new slot scheduling takes time. The mobile nodes interfere with other nodes for a while in between. It strongly affects the quality of service (QoS) of a running application if it should be continuously performed for highly mobile nodes.

M-LMAC [98] supports mobility of singular nodes with event-based or very low sampling rate data generation. Static nodes use the LMAC mechanism for their slot scheduling. Mobile nodes that are not synchronized to the static nodes, in contrast, use carrier sensing whenever they have data to transmit. The time slots assigned to the static nodes are doubled in length to include a contention-based period. Every static node sends an Announcement Message (AM) at the beginning of the contention period of its assigned time slot and starts listening to the channel to receive possible responses from a mobile node. In the case that a mobile node in the neighborhood has data to transmit, it scans the channel to find an AM. If so, the mobile node tries to transmit its data by CSMA. If the data transmission succeeds, the static node sends an acknowledgement to the transmitting mobile node. This protocol is well-suited for applications with sparse mobile node distributions, having infrequent data transmissions, such as a fire fighting application (as mentioned and studied by the authors). However, it is not the best choice for applications like health monitoring, in which several clusters of mobile nodes exist with data transmission in every TDMA frame. In such applications, all nodes in a cluster (WBAN) are always in each other's interference range. Therefore, M-LMAC causes very many collisions that lead to a degraded performance. Moreover, a static node is only able to receive data from at most one mobile node in each frame, even if no collisions occur. This further limits the services for important body-related data in health applications. Moreover, it imposes high listening energy consumption overhead to the mobile nodes to scan the channel for AM. In a health monitoring application, in particular, body sensor nodes are very power-constrained nodes.

MMAC [12] provides a proactive rescheduling mechanism with the slot scheduling principles of TRAMA [73] as its core. A localization service is used to predict the mobility pattern of the mobile nodes. According to the predicted location of the nodes, the schedules are adapted. This way, the protocol tries to make itself ready for future movements. However, besides the complexity of the localization service and predictions, the performance of the protocol strongly depends on the accuracy of the predictions.

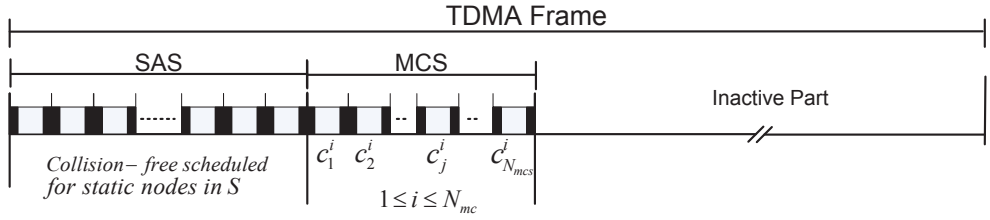


Figure 4.1: A TDMA frame containing the SAS and MCS parts.

Although the presented solutions for node mobility in TDMA-based MAC protocols improve performance of such protocols, none of them has specifically been designed for supporting group (cluster) mobility of nodes (like a WBAN). In this sense, MCMAC is the first work that addresses cluster mobility in WSNs. Idle listening to the mobile nodes imposes a high power consumption overhead and methods to avoid it are worth considering. This is also not addressed properly in the current protocols. These issues are addressed by MCMAC.

4.3 Supporting Cluster Mobility

The MCMAC protocol uses a hybrid schedule-based and contention-based communication mechanism to support cluster mobility. In this section, the MCMAC mechanism and its different variations are presented.

4.3.1 A Hybrid Communication Approach

In the base TDMA-based MAC layer explained in Chapter 2 (Fig. 2.2), slots in the active part of the frames are exclusively assigned to different sensor nodes in a neighborhood. In MCMAC, the slots that are used by static nodes and mobile cluster nodes are separated and different accessing paradigms are used for each part. The active part of the frames is split into two separate parts shown in Fig. 4.1, a static active section (SAS) and a mobile cluster section (MCS). The SAS is used by the static nodes to transmit their data. A pure TDMA scheme with the distributed dynamic slot scheduling strategy of LMAC [97] is exploited in this part. Using this approach we make benefit from spatial reuse of RF channels. So each static node occupies a slot from SAS that is unique in its two hop neighborhood. However, other scheduling mechanisms can also be used here without impacting the other part of the protocol. In any case, these mechanisms try to minimize the number of SAS slots (N_{sas}) through smart slot reuse. Static nodes exchange data with each other by transmitting packets in their scheduled slot and listening to other SAS slots.

A contention-based mechanism is exploited within the TDMA slots of the second part of the frame (MCS). Each node c_j^i in cluster C^i is assigned statically a unique slot in this

part to transmit its data. The size of the MCS part (N_{mcs}) is then determined by the number of nodes in the mobile clusters. The maximum cluster size is taken as the MCS length. Nodes within a cluster can always access the channel without interfering with each other and without the need for rescheduling upon intra-cluster topology changes due to individual node movements. Given the fact that the nodes in a cluster are mostly within a one- or two-hop neighborhood of each other, statically assigning a unique slot to each node of a cluster is an efficient approach. Considering the high mobility of the nodes within the cluster, that leads to frequent cluster topology changes, adaptive slot reuse in this part is inefficient and costly. In Sec. 4.6, where we generalize the protocol to support high cluster densities, the mobile clusters are classified taking their size into account in the case that the sizes of different clusters in the network vary considerably. Multiple MCS parts of different sizes are then used.

The sensor nodes from different mobile clusters share a single MCS part. For instance, node c_j^i from cluster C^i and node c_j^k from cluster C^k share the j^{th} MCS slot (for simplicity, we assume that the nodes in the cluster are indexed according to their assigned slots). Thus, when these two clusters are in each other's range, they compete to access the same slot and transmit their data. Fig. 4.1 illustrates the structure of a frame and slot allocations in the MCMAC protocol. A contention-based mechanism is used for accessing such TDMA slot. Two variations of the protocol are developed using CSMA or slotted ALOHA as the contention-based paradigms. The appropriate version is then selected according to the circumstances and hardware facilities in the application. Both variations and guidelines for decision making are presented in detail in this section.

Static nodes listen in the MCS slots of the frame to receive data from cluster nodes and forward it to other static nodes to eventually be delivered to the sink node. As the position of the MCS is known in the frame, the static nodes do not need to listen in the whole inactive part of the frame to possibly receive packets from mobile clusters. This is the main reason for specifying a fixed MCS in the frames with a contention-based access by the mobile cluster nodes. Another important advantage is that by developing a smart listening schedule, static nodes can avoid listening in MCS slots when no mobile cluster is around. We present such a mechanism later in this chapter. According to the application scenario, cluster nodes may also listen to the SAS to receive information from the static network. Feedbacks, comments, and alarms for the person wearing the WBAN are examples of such information.

4.3.2 Carrier Sense Multiple Access Paradigm

CSMA is used as the first contention-based mechanism for accessing the shared MCS slots. Every cluster node performs carrier sensing before transmitting in its dedicated slot. The duration of a time slot is extended by the duration $T_{cp} + T_{sw}$, where T_{cp} is the length of the *contention period* in which nodes compete to transmit their data. T_{sw} stands for the switching time of the radio transceiver from carrier sensing to transmission mode. Fig. 4.2 shows the structure of one time slot prepared for performing CSMA. $T_{transmit}$ stands for the actual time required for transmitting one fixed-sized packet. Cluster nodes randomly pick a time point (t_r) in the contention period and sense the channel form the

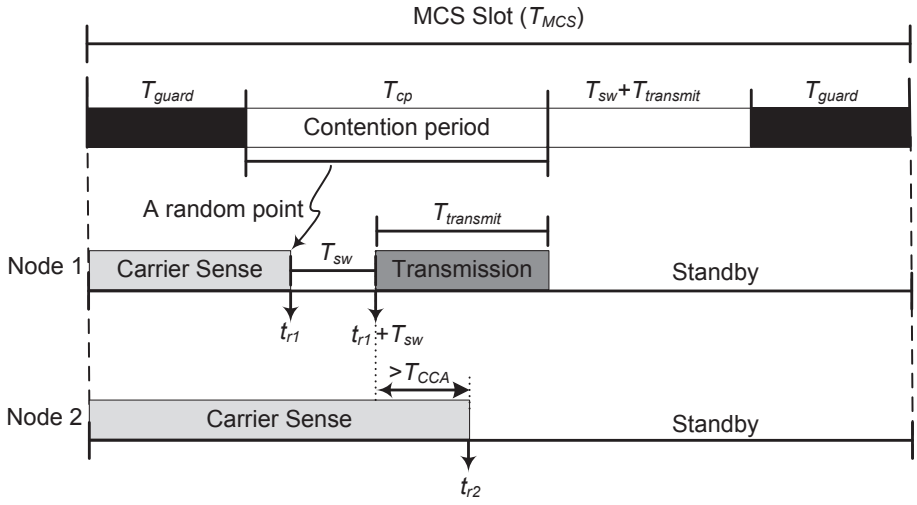


Figure 4.2: The structure of a time slot prepared for performing CSMA in the MCS part of the frame.

beginning of the slot till t_r. If the channel is found to be idle, the node switches to transmission mode to start sending its packet (the case for Node 1 in Fig. 4.2). If the node detects a carrier, it waits for the next frame to try again (Node 2). Note that T_{CCA} is the clear channel assessment (CCA) time, which is the minimum time that the radio chip requires to sense the wave to be able to decide on the existence of a carrier on the channel. For instance, the value of T_{CCA} in both the Chipcon CC2420 [6] and Nordic nRF24L01 [4] transceivers is $128\mu s$ (8 symbol periods).

There are several aspects that should be considered for performing CSMA in MCS slots. The features of the carrier sensing facilities of the radio transceiver and the TDMA framing and slot length are important factors. To obtain a better understanding how these factors influence the performance of CSMA, we analyse the probability of collision or successful transmission in an MCS slot. Assume that there are two nodes in each other's interference range. To have a successful transmission of one of them, the distance of the selected random points within the contention period should be at least (T_{CCA}+T_{sw}) apart. Suppose without loss of generality that in one round $t_{r1} < t_{r2}$. The real transmission of Node 1 starts at time $t_{r1} + T_{sw}$. Node 2 should at least sense the signal of the first node for a duration of T_{CCA} to detect that. The probability of a transmission without collision is as follows.

$$P_{\overline{col}}(Node1, Node2) = P\big(|t_{r1} - t_{r2}| > T_{sw} + T_{CCA}\big) \tag{4.1}$$

Points t_{r1} and t_{r2} are selections out of two independent uniform distributions from the contention period of length T_{cp}. In the following, Eqn. 4.1 is split into two parts for two different cases.

$$\begin{aligned} P_{\overline{col}}(Node1, Node2) &= P\big(t_{r1} > (t_{r2} + T_{sw} + T_{CCA}) \mid t_{r1} > t_{r2}\big) + \\ &\quad P\big(t_{r2} > (t_{r1} + T_{sw} + T_{CCA}) \mid t_{r2} > t_{r1}\big) \\ &= 2 \times \frac{\big(T_{cp} - (T_{sw} + T_{CCA})\big)^2/2}{{T_{cp}}^2} \end{aligned} \tag{4.2}$$

Note that the whole selection area for these two random variables is a square of size T_{cp}^2. In each case, the interesting part of the square that satisfies the requirement for avoiding collision is of size $\big(T_{cp} - (T_{sw} + T_{CCA})\big)^2/2$. Therefore, we get Eqn. 4.3 to be the probability of a collision-free transmission of one of these two nodes.

$$P_{\overline{col}}(Node1, Node2) = \left(1 - \frac{T_{sw} + T_{CCA}}{T_{cp}}\right)^2 \tag{4.3}$$

Note that the effect of the hidden terminal problem is not taken into account in this calculation.

According to the protocol, the i^{th} node from all mobile clusters share the same MCS slots (the i^{th} slot). Let Γ be the set of mobile clusters that are in each other's interference range at a certain moment. The size of this set ($\gamma = |\Gamma|$) determines the the number of mobile clusters that compete for transmission in the same MCS slot. To have a successful transmission by one mobile node in this situation, all random time points selected by mobile nodes but the earliest one should be at least $T_{sw} + T_{CCA}$ later than the earliest one. In this case, the earliest node succeeds to transmit without colliding with other nodes. As different nodes select their t_r points independently, the probability of not colliding with $\gamma - 1$ nodes is the product of the probability of not colliding with each of them (Eqn. 4.3). To which node the earliest point belongs does not make a difference. Now, from the perspective of one competing cluster node c_j^i, the probability of selecting the earliest point ($1/\gamma$) and successfully transmitting (winning the lottery and no collision happening) in the current round is as follows.

$$\begin{aligned} P_{CSMA}(c_j^i) &= \frac{1}{\gamma}\left(\prod_{c_j^k : C^k \in \Gamma, k \neq i} P_{\overline{col}}(c_j^i, c_j^k)\right) \\ &= \frac{1}{\gamma}\left(1 - \frac{T_{sw} + T_{CCA}}{T_{cp}}\right)^{2(\gamma-1)} \qquad 1 \leq \gamma \leq N_{mc} \end{aligned} \tag{4.4}$$

A longer contention period leads to a higher chance of success. The appropriate value should be set considering the expected success probability (QoS requirements), the tendency of mobile clusters to gather in the target application, and the radio chip features (T_{CCA} and T_{sw}). $T_{transmit}$ is also an important factor. A long contention period is not reasonable for very short transmissions, as it imposes a high overhead.

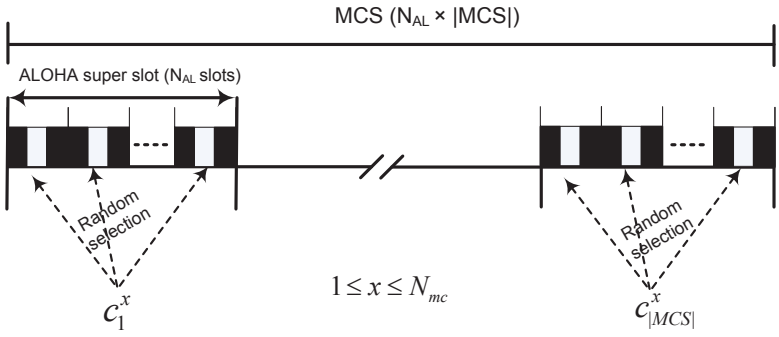


Figure 4.3: The structure of a super slot prepared for performing slotted ALOHA in the MCS part of the frame.

4.3.3 Slotted ALOHA

The second contention-based scheme for accessing MCS slots is a slotted ALOHA mechanism. In this version, N_{AL} slots construct one *ALOHA super slot* and are dedicated to the j^{th} node of all clusters (instead of one single slot). The number of super slots in an MCS is then N_{mcs}. Mobile cluster nodes in each round randomly select one slot among N_{AL} in their ALOHA super slot and transmit their data in that slot. Fig. 4.3 depicts the structure of the MCS part of the frame and its ALOHA super slots. No carrier sensing is performed by the mobile cluster nodes. Moreover, the structure of the time slots does not change (Fig. 2.2). It makes the implementation of the protocol simpler than the CSMA approach.

Assume that in some frame, γ sensor nodes from different mobile clusters are in each other's range and want to transmit in the same ALOHA super slot. This means that each of the γ nodes selects a random slot out of N_{AL} slots. In such a case, a certain cluster node (c_j^i) successfully transmits if all other nodes ($c_j^k : C^k \in \Gamma$) select another slot than the one selected by that node. The probability that node c_j^i selects a certain slot is $\frac{1}{N_{AL}}$. It succeeds to transmit only if none of the other $\gamma - 1$ competing nodes select the same slot. This happens with the probability of $(1 - \frac{1}{N_{AL}})^{\gamma-1}$. Such competition may also occur for transmission in other slots out of N_{AL} slots. Therefore, the probability of a collision-free transmission for the given node (c_j^i) is calculated as follows.

$$\begin{aligned} P_{ALOHA}(c_j^i) &= N_{AL} \times \frac{1}{N_{AL}} \times \left(1 - \frac{1}{N_{AL}}\right)^{\gamma-1} \\ &= \left(1 - \frac{1}{N_{AL}}\right)^{\gamma-1} \quad 1 \leq \gamma \leq N_{mc} \end{aligned} \tag{4.5}$$

The controllable parameter here is the number of slots in an ALOHA super slots (N_{AL}). According to Eqn. 4.5, a bigger value of N_{AL} increases the chance of success. Again the value of γ varies by movement of the clusters and depends on the tendency of mobile clusters to gather in the target application. Notice that there is a difference with respect to the successful transmissions between these two contention-based approaches. In CSMA at most one node may successfully transmit its data in each frame. In slotted ALOHA however, it may happen that multiple nodes are successful in one TDMA frame and do not collide with other nodes in the range.

4.3.4 Guidelines for Decision Making

Several factors should be considered to decide which contention-based approach fits better for a certain application. Slotted ALOHA is easier to implement and does not need special support, like carrier sensing, from the radio chip. When the transmission time ($T_{transmit}$) and accordingly slot length (T_{slot}) are short, this approach is indeed worth considering. On the other hand, CSMA takes more implementation efforts and the features of the radio chip (like duration of carrier sensing and switching time) influence the efficiency of the protocol. However, for applications with longer transmissions (either longer packets or lower data rates), CSMA performs better. This is because we need to have a rather long contention period for performing carrier sensing. Having a short transmission after a long contention period is not an efficient approach.

To have a quantitative comparison between these two approaches in an application, we investigate the amount of listening activities imposed to static nodes to listen to the MCS part. Assume that $T_{slot} = T_{transmit} + 2T_{guard}$ is the length of a basic slot (Fig. 2.2) containing two guard times and one transmission time. The time duration of the MCS part of the frame (T_{MCS}) is given by Eqn. 4.6 and Eqn. 4.7 for CSMA and ALOHA, respectively. This is the time in which static nodes listen to the channel to possibly receive from the cluster nodes. For CSMA, we assume that the static node listen to the channel for the whole slot duration.

$$T_{MCS}(CSMA) = [\ T_{slot} + T_{sw} + T_{cp}] \times N_{mcs} \tag{4.6}$$

$$T_{MCS}(ALOHA) = T_{slot} \times N_{AL} \times N_{mcs} \tag{4.7}$$

Comparing $T_{MCS}(CSMA)$ and $T_{MCS}(ALOHA)$ reveals which approach fits better in the target application. This comparison should be done when the parameters of the CSMA and ALOHA are set in such a way that both provide the same transmission success probability for a given cluster node in the same situation. T_{cp} and N_{AL} are controllable parameters of the CSMA and slotted ALOHA mechanisms and are determined by solving Eqn. 4.4 and Eqn. 4.5 for a given probability of success, and a given number of gathered nodes by Eqn. 4.8 and Eqn. 4.9, respectively.

$$T_{cp} = (T_{sw} + T_{CCA}).\left(1 - [P_{CSMA}.\gamma]^{\frac{1}{2(\gamma-1)}}\right)^{-1} \quad \gamma > 1 \tag{4.8}$$

Table 4.1: Values of the parameters in two different deployments.

Parameter	*MyriaNed [95]*	*Mica2/Z-MAC [78]*
Data rate	2 Mbps	19.2 Kbps
T_{CCA}	128 μs	400 μs
T_{sw}	130 μs	200 μs
Slot length (T_{slot})	764 μs	50 ms
T_{cp} (Eqn. 4.8)	2500 μs	6 ms
N_{AL} (Eqn. 4.9)	2	2
$T_{MCS}(CSMA)$ (Eqn. 4.6)	3400 μs $\times N_{mcs}$	56 ms $\times N_{mcs}$
$T_{MCS}(ALOHA)$ (Eqn. 4.7)	1528 μs $\times N_{mcs}$	100 ms $\times N_{mcs}$

$$N_{AL} = \left\lceil \left(1 - [P_{ALOHA}]^{\frac{1}{\gamma-1}}\right)^{-1} \right\rceil \quad \gamma > 1 \tag{4.9}$$

The value of γ represents the number of mobile clusters that are in each other's interference range and its value may have a certain distribution over the network runtime in a specific deployment. According to the application scenario, the maximum value, the median of the distribution, or a weighted average of the instantaneous values of γ can be put in the equations. For instance, if the data delivery performance in the worst-case (gathering of all clusters) is an application requirement, the maximum value, which is the number of mobile clusters in the network, can be applied. For the success probabilities, equal values are set ($P_{CSMA} = P_{ALOHA}$), so that both approaches are compared while providing the same transmit success probability. The required probability depends on the expected QoS in the running application, and the retransmission opportunities of the mobile cluster node. Inserting the values coming out from Eqn. 4.8 and Eqn. 4.9 in Eqn. 4.6 and Eqn. 4.7, a fair comparison can be done comparing the duration of the MCS part in these two mechanisms.

An example illustrates this procedure. We consider two deployments using different wireless nodes and TDMA structures. Assume that MyriaNed nodes [95] are used in the first setting (explained in Chapter 2). Each slot contains one packet and leads to small slots (taking the high data rate into account). In contrast, consider Z-MAC [78] settings using Mica2 nodes. The data rate is much lower and so the slot length is much bigger. The values of different parameters of these two cases are shown in Table 4.1 together with the results of the calculations. In this calculation, we assume that $\gamma = 2$ and the required probability of successful transmission is 40%[1]

Comparing the calculated value for $T_{MCS}(CSMA)$ and $T_{MCS}(ALOHA)$ allows us to decide which approach fits better for a setting. The results clearly show that in the first case using MyriaNed nodes, the slotted ALOHA version performs better as $T_{MCS}(CSMA) \gg T_{MCS}(ALOHA)$. The reason is indeed the small size of the slots. In contrast, the CSMA mechanism is surely promising for the second case because $T_{MCS}(CSMA) \ll T_{MCS}(ALOHA)$.

[1] In Eqn. 4.4, the maximum value of the successful transmission probability is $\frac{1}{\gamma}$, when the contention period is infinitely large. Since we assume $\gamma = 2$ in our calculations, the maximum is 50%. Because of that, we use a low value (e.g., 40%) for the required successful transmission in this example.

4.4 Scheduling the Listening to Mobile Clusters

Idle listening is one of the power wasting activities in wireless communications and is to be minimized. In the MCMAC protocol, every static node listens to the MCS slots to receive data from sensor nodes in the mobile clusters and route it to the sink nodes. However, when no cluster is around, the static node can stop listening to the MCS part to save energy. We fortify the MCMAC protocol with a mechanism with which the static nodes schedule their listening to mobile clusters. One of the reasons for having a shared contention-based part (MCS) in each frame is the possibility to minimize such idle listening when no cluster is around the static nodes.

The mechanism works as follows. Each static node continuously estimates the hop-distance of the nearest cluster to itself. By following the variations of the estimated hop-distance, the node tries to track the movement of the clusters. Based on that, the node decides how often it needs to listen to the MCS slots of the frame. The objective is to reduce idle listening while avoiding data losses due to not listening to the clusters in the range. Both our experiments and simulations show a very considerable energy saving (around 70% on average in our experiments) for static nodes without negatively influencing other performance metrics such as latency and delivery ratio. Clearly, the number of mobile clusters, its density, and the mobility pattern in a deployment play an important role in the energy saving.

4.4.1 Hop-Distance Estimation

The goal is to have a distributed mechanism with which every static node estimates how far the clusters are away from it. As there is only one MCS part in the frame, static nodes only need to estimate the hop-distance (d) to the nearest cluster. Each node in the network propagates its estimated distance by adding it to its packets. Cluster nodes always send zero ($d = 0$). Every static node that receives zero in a round, realizes that there is at least one cluster in its one-hop neighborhood. Further, each node receives the hop-distance of its direct neighbors in each round. It then estimates its own distance as the minimum distance of its neighbors plus one. In the case that a node did not receive any packet from any neighbor in a round, it keeps its estimate of the previous round. Fig. 4.4 shows the value of d for all nodes in a given 6×5 static mesh network with two mobile clusters.

Note that the static nodes need to know how far the mobile clusters are to be able to schedule their listening. When the mobile clusters are far a way, the node can relax from listening to the MCS part. Any *big* value of the hop-distance has almost the same meaning; taking this into consideration we can specify a maximum value for the hop-distance (d_{max}). It allows a limited small space for transmitting its value. A value $d_{max} = 8$ is big enough for a typical network size. Note that we do not need to have an exact estimation of hop-distance, especially when its value is not low. Therefore, four bits is enough to transmit the parameter $0 \leq d \leq 8$. In comparison with the typical packet length (e.g., 32 bytes), the overhead of integrating hop-distance d to the packets is low. Note that the hop-distance value is added to the data packets in protocols with

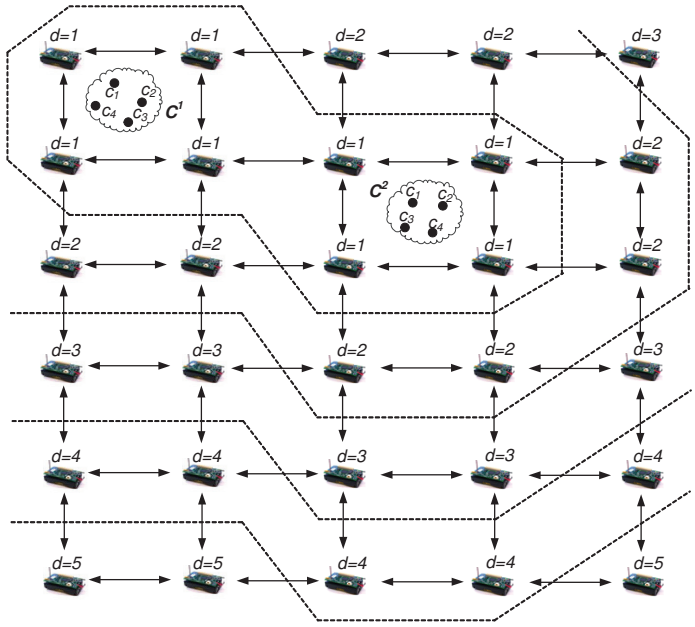


Figure 4.4: Hop-distance estimation to the nearest cluster in a 6×5 nodes static WSN with two mobile clusters, each containing four nodes. Regions with the same d values are pointed out. The regular deployment is only to illustrate the hop-distance concept. The MCMAC protocol does not assume such a deployment.

frequent periodic data exchange. In other protocols, very small control packets may be added. In TDMA-based protocols, frequent data or control packets already exist for other purposes, in particular time synchronization.

The minimum hop-distance to the mobile clusters is considered as an estimator of physical distance of the nearest mobile cluster to a certain static node. However, translation of the hop-distance to the physical distance is not straightforward. Many issues play a role here. Non-uniform transmission patterns and heterogeneous transmit ranges of the sensor nodes, spatial and temporal link quality variations, and available paths for movements are influencing the precision of the estimations. In particular, instantaneous connections and disconnections make the hop-distance d vary over time even when no cluster is moving. Fig. 4.5 shows the estimated hop-distance for a sample static node in one of our experiments for a certain period of time. The general variation pattern of d reveals that the cluster goes far away and then approaches the node. However, the fluctuations due to wireless link variations are quite visible. Taking such issues into account, a conservative approach together with some averaging methods seems needed for scheduling the listening activities of the static nodes. We discuss such an approach next.

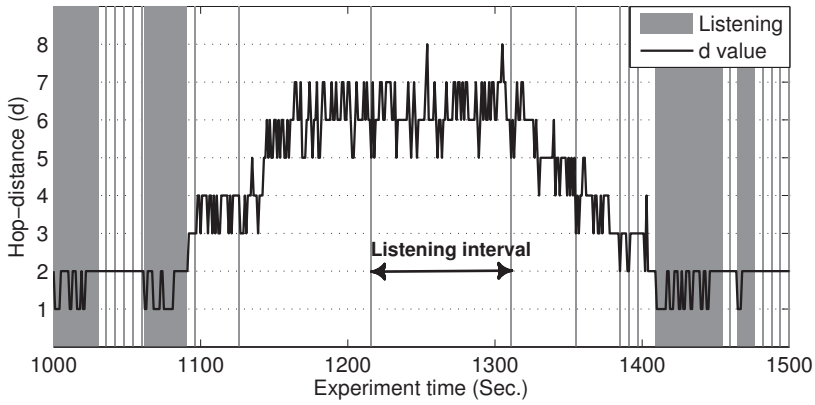


Figure 4.5: The variations of the estimated hop-distance d in a duration of a real experiment for a sample static node and its decision about listening to the MCS part.

4.4.2 Listening Schedule to Mobile Clusters

The function StaticOpt() in Algorithm 2 decides whether or not the node listens to the MCS part. Algorithm 3 presents this function. Every node maintains a listening interval time T_l that determines how often the static node listens to the MCS part of the frame. It is adjusted considering the value of parameter d and its history record. The node listens to the MCS part (when StaticOpt() returns *true*) for one round in every T_l round(s). Fig. 4.5 shows the listening interval of a static node in our experiments for a certain time duration and the decision of the static node about listening to the MCS part. The process for adjusting the listening interval should balance two aspects. First, it should minimize idle listening when there is no mobile cluster around. Second, it should minimize the packet losses caused by not listening to the MCS part, when there is a cluster in the neighborhood.

An Additive Increase Multiplicative Decrease (AIMD) [19, 52] mechanism is exploited for adjusting the value of T_l. In the AIMD mechanism, the growth of the time interval parameter is done linearly, but an exponential decrease is used when it is necessary to decrease the parameter. Thus, AIMD provides a conservative approach for adjusting the value of T_l by a fast reduction of the time interval size when a mobile cluster is approaching, and a slow increase when all clusters around the static node are moving away. To perform the AIMD, different aspects of the hop-distance d are considered; the last estimated value $d(t)$, an average over a limited history $d_{avg}(t)$, its last variation $\Delta d(t)$, and the deviation from the average $\delta_d(t)$. To reduce the effect of variations of d due to the link variations and to separate this effect from a real change of the value of d caused by the movement of a cluster, we use an exponentially weighted average over a limited history (of H TDMA rounds) of the parameter d (Eqn. 4.10). The parameter α determines the gain of the weighted averaging.

Algorithm 2: Behavior of a static node.

```
1  for every TDMA frame t do
2      Lst ← StaticOpt(d(t));
3      Rxd[i] = d_max + 1,  0 ≤ i ≤ N_sas;                    /* reset values */
4      for i = 1 to N_sas do
5          if i ≠ ownTxSlot then
6              RxPacket ← Listen() ;
7              if Received any then Rxd[i] = RxPacket.d;
8          else
9              TxPacket.d ← d(t);
10             Transmit(TxPacket);
11         end
12     end
13     if Lst then
14         for i = 1 to N_mcs do
15             RxPacket ← Listen() ;
16             if Received any then Rxd[0] ← 0;
17         end
18     end
19     d ← min{ Rxd[i] | i : 0 ≤ i ≤ N_sas } + 1;
20     if d ≤ d_max then
21         d(t + 1) ← d ;
22     else
23         d(t + 1) ← d(t) ;
24     end
25     Go to sleep mode until the next frame
26 end
```

$$d_{avg}(t) = \left\lfloor \frac{\sum_{k=1}^{H} a_k \cdot d(t-k)}{\sum_{k=1}^{H} a_k} \right\rfloor, a_k = (\frac{1}{\alpha})^k \ , \alpha \geq 1 \tag{4.10}$$

$$\delta_d(t) = d(t) - d_{avg}(t) \tag{4.11}$$

$$\Delta d(t) = d(t) - d(t-1) \tag{4.12}$$

These parameters are used in Eqn. 4.13 to adjust the value of the listening interval. T_l^{t-1} is the current listening interval. T_l' is an intermediate version of the next value of the listening interval based on the AIMD paradigm. The method for adjusting the time interval is developed based on several experiments in various circumstances taking the observed variations of parameter d over time into account. The first two cases in Eqn. 4.13 provide an additive increase of the listening interval when the nearest cluster is estimated to go farther or remain in the same place. While the hop-distance remains fixed, it adds one step (TDMA round). When the value of d increases, it can increase with bigger steps. We set such value as a factor (β) of the current hop-distance. Farther clusters allow us to increase the interval with bigger steps without any risk. For the current implementation we set $\beta = 1$.

Algorithm 3: The optimization function.

```
1  StaticOpt()
   Input: d(t)                                   /* the last estimated hop-distance */
   Output: Lst                                   /* if node listens to the MCS part */
2  calculate d_avg(t), δ_d(t), and Δd(t) using Eqn. 4.10 − 4.12 ;
3  switch Δd(t) do
4  |  case Δd(t) > 0   T_l^t ← T_l^{t−1} + d(t);
5  |  case Δd(t) = 0   T_l^t ← T_l^{t−1} + 1;
6  |  case Δd(t) < 0   T_l^t ← ⌊ T_l^{t−1} / 2^{|δ_d(t)|} ⌋;
7  endsw
8  if T_l^t > T_{l,max}[d_avg(t)] then T_l^t ← T_{l,max}[d_avg(t)]
9  IntervalCnt ← IntervalCnt + 1;
10 if IntervalCnt ≥ T_l^t then
11 |  Lst ← true;
12 |  IntervalCnt ← 0;
13 else
14 |  Lst ← false;
15 end
```

$$T_l' = \begin{cases} T_l^{t-1} + \beta \cdot d(t) & \Delta d(t) > 0 \\ T_l^{t-1} + 1 & \Delta d(t) = 0 \\ \left\lfloor \frac{T_l^{t-1}}{2^{|\delta_d(t)|}} \right\rfloor & \Delta d(t) < 0 \end{cases} \tag{4.13}$$

The last case in Eqn. 4.13 performs the multiplicative decrease when d steps down. In this case, the value of the listening interval is divided by an exponential factor determined by the deviation of the hop-distance to its average value ($2^{|\delta_d|}$).

However, the listening interval should not be limitless. Consider a situation in which the cluster nodes stay fixed for a very long period of time. The listening interval additively increases. At a time, a mobile cluster starts moving. If T_l has become too big, decreasing that takes time even with the multiplicative method. Joining new mobile clusters is another issue that should be considered here. Thus a maximum allowed value is set for the listening interval individually determined for each hop-distance average. $T_{l,max}[k]$ is the length limit of the listening interval when the weighted average hop-distance is k ($d_{avg}(t) = k$).

$$T_l^t = min\{\ T_l'\ ,\ T_{l,max}\,[d_{avg}(t)]\ \} \tag{4.14}$$

T_l^t is the new value of the listening interval. $T_{l,max}[1]$ is always set to one so that a node listens to the *MCS* part in all rounds when there is a cluster around. The proper maximum value for other cases depends on the density of the network and the average speed of the mobile clusters in the application.

4.4.3 Trade-offs: An Empirical Exploration

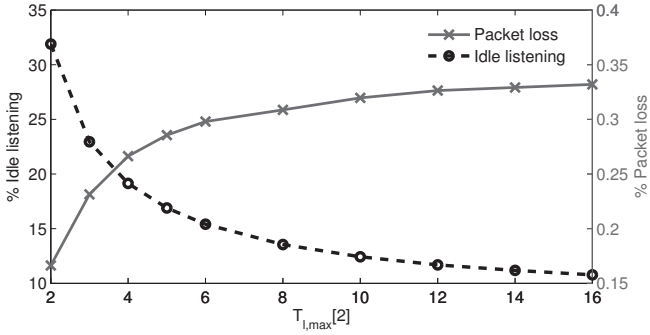
The presented listening optimization mechanism has a couple of controllable parameters. In particular, the length of the averaging history (H), the gain of the weighted averaging (α), and the maximum allowed limits for the listening interval ($T_{l,max}[]$) are parameter settings that influence the performance of the mechanism.

To get insight in the variations of parameter d in a real-life network, we performed 6 hours of experiments with the setup explained in Sec. 4.5 with only one mobile cluster showing various mobility behavior. During the experiment, the value of hop-distance d and packet reception are logged in 60 static nodes. In this experiment, static nodes were always listening to the MCS part. By off-line mimicking of the listening scheduling mechanism with various configurations using the logged data (hop-distance and packet reception in each round), we can evaluate the efficiency with different parameter settings. In particular, we considered parameters H, α, and $T_{l,max}[2]$. Ten different values were set for each parameter. Thus, in total, 1000 configurations have been evaluated. $T_{l,max}[2]$ is specifically considered because experiments show that its value has an important influence on the performance of the mechanism. $T_{l,max}[2]$ is the maximum allowed size of the listening interval when the static node is not directly in the range of a mobile cluster, but it is within 2-hop distance from a cluster. It means that there is always a chance that the cluster moves closer toward the static node in a short time. In this case, a big maximum value delays listening to the mobile cluster. On the other hand, a small value of $T_{l,max}[2]$ imposes a high idle listening overhead because the cluster may remain in the 2-hop neighborhood for a long time.

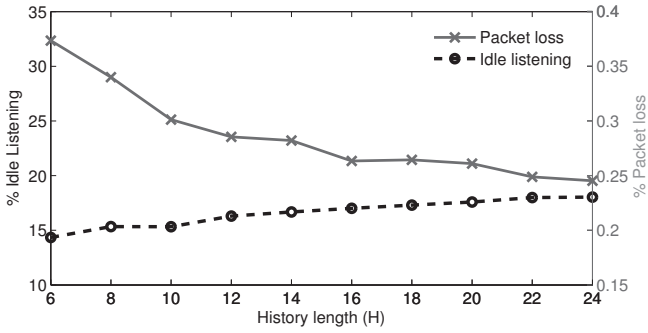
The observed performance metrics are the percentage of rounds with idle listening to the MCS part and packet losses due to non-listening. When the off-line evaluation of function StaticOpt() returns *false* and a packet is received from a mobile cluster in the real experiment in that specific round, it is counted as a packet loss. Idle listening happens whenever StaticOpt() returns *true*, but no packet is received in the MCS part. As the obtained metrics vary over different static nodes, we calculate the average metrics over all static nodes.

Fig. 4.6 depicts the explored trade-offs between the idle listening and packet losses in term of the parameters of the mechanism. To investigate the influence of a parameter on the performance of the mechanism, in each iteration, various values for a parameter are tried while other parameters are fixed. Fig. 4.6(a) shows the trade-off in terms of the maximum allowed listening interval when the average hop-distance to the nearest mobile cluster is 2 ($T_{l,max}[2]$). Increasing the value of this parameter strongly decreases the idle listening at the cost of more packet losses from mobile cluster nodes and vice versa. Note that other limits of the listening interval ($T_{l,max}[n]$, $n > 2$) also make the same trade-off, but with a smaller influence on the performance metrics.

Fig. 4.6(b) shows the trade-off in terms of the length of the averaging history (H). When a longer record is taken into account, the idle listening increases because the additive increase of the listening interval is triggered with more delay. However, the influence of history length on performance metrics is not as strong as the one from $T_{l,max}[2]$. Exploring the effect of α gives the same conclusions. Considering the averaging



(a) Trade-off made by maximum size of listening interval.

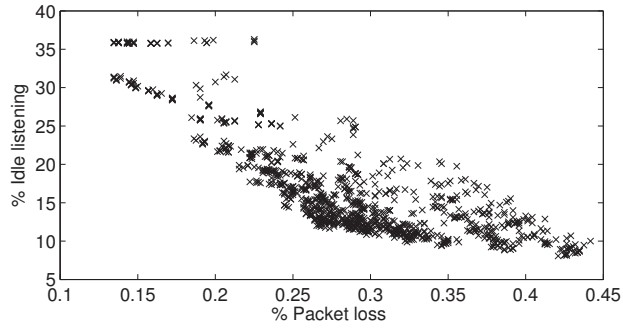


(b) Trade-off in terms of the averaging length (H).

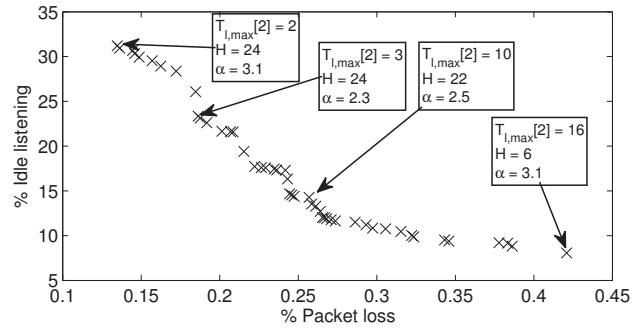
Figure 4.6: Trade-off between idle listening to MCS part and packet losses in terms of the parameters of the listening scheduling.

parameters (H and α), the conclusion is that using a smoother averaging (greater H and smaller α) so that the new values of d have softer influence on $d_{avg}(t)$, a lower idle listening is achieved. In such case, the reaction of the mechanism is delayed when d drops, and so more packets are lost. If the averaging is set to be more dependent on the new values (smaller H and greater α), packet losses decrease and idle listening increases. This reflects the behavior of the AIMD mechanism.

Fig. 4.7(a) depicts the average obtained metrics for all tried configurations. This clearly shows that some configurations are far from optimal. Fig. 4.7(b) shows the *Pareto* points among all configurations. These are the settings that are not dominated by any other setting. A Pareto configuration should be finally selected based on the QoS constraints in the application scenario. When packet losses cannot be tolerated, the settings on the top left side of the graph are selected. Points on the other extreme are appropriate when energy saving is an important requirement. Notice that a packet loss measured in a static node would not necessarily mean the loss of a packet from the cluster node.



(a) All explored configurations



(b) Pareto point configurations and some instance settings

Figure 4.7: Empirical exploration of the configuration space for listening scheduling mechanism and obtained metrics.

According to the network architecture, there can be other static nodes in the range that listen to the MCS part and so receive the packets. Taking that into account, we can say that packet losses might be tolerated to some extent.

4.5 Performance Evaluation

4.5.1 Experimental Setup

We investigate the applicability and performance of the cluster mobility support through various real-world experiments and simulations. During the real-world experiments, we used a rather large-scale indoor deployment using wireless nodes. Real human behavior in a working environment, such as sitting in the office, walking, and gathering of several people wearing the nodes (mobile clusters), makes the experiments realistic. Various performance metrics are investigated during these experiments through detailed analysis

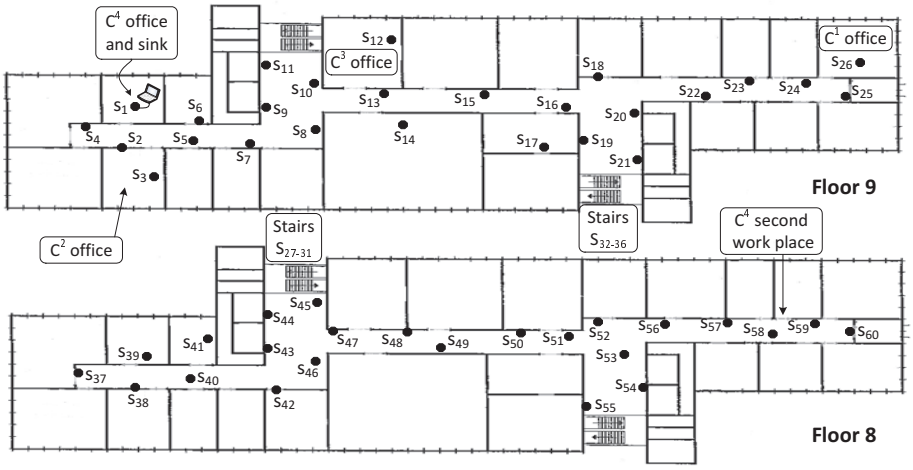


Figure 4.8: Node deployment on two floors (8^{th} and 9^{th}) of the Electrical Engineering department of TU/e.

of the logged data. Besides the experiments, we also extensively simulated the protocol in several network setups. We run different simulations with and without performing listening scheduling to evaluate its influence on various QoS metrics. We also simulated network setups with various scales and cluster densities.

In the MAC layer of the protocol stack used in the experiments and simulations, slots in the SAS part of the TDMA frames are assigned to the static nodes based on the slot scheduling of the LMAC protocol [97]. Each static node occupies a time slot to send its data packets that is unique in the node's two-hop neighborhood. Therefore, the communication between static nodes is in principle contention-free. Nodes synchronize using a decentralized frame synchronization method as explained in Chapter 2.

The static nodes use the default data dissemination mechanism of the MyriaNed protocol stack (Chapter 2) to propagate data items generated by static nodes themselves as well as data received from mobile cluster nodes. On top of the data routing protocol, a monitoring application is used in which all sensor nodes periodically sense some parameter and pass the sensed data to the lower layer. The physical parameter being sensed is not relevant for the experiment.

4.5.2 Real-World Experiments

Deployment setup: We deployed a WSN testbed on two floors of the Electrical Engineering department of the TU/e, which covers an area of around $2400m^2$. In total 76 MyriaNed wireless nodes are used from which 60 nodes are static nodes and 16 nodes form four mobile clusters each having four nodes. Fig. 4.8 shows the node placement in our testbed. Although there are some (low-quality) direct links between nodes on

different floors, we deployed several nodes in the stair ways in both ends of the floor to ensure a reliable network connectivity. Four volunteers took part in the experiments to carry cluster nodes. The figure shows the working offices of the volunteers where they mostly sit and work. They perform their normal movement behavior such as meetings and coffee breaks. Besides that, they were asked to walk through the network (including the other floor) for several times during each experiment to ensure a minimum level of cluster mobility. Node s_1 is the sink node and is connected to a laptop. Other static nodes are between 1 (e.g., s_2) to 7 (e.g., s_{60}) hops away from the sink node. Each node transmits one packet in every frame.

Performed experiments: We performed four different experiments each lasting three hours. The limiting factor to the duration of each experiment is the memory capacity of the MyriaNed nodes for logging network activities. After each experiment, all nodes are read to gather logged information. The first two experiments are done with only one mobile cluster and without performing listening scheduling. The goal of these experiments was to log hop-distance of the mobile cluster to the static nodes (d) as well as other network data to explore the trade-offs in the listening scheduling mechanism. The result was discussed in Sec. 4.4.3. The later two experiments were done with four mobile clusters and with the full version of the protocol. Required information is logged in all nodes as well as the laptop connected to the sink to compute different performance metrics.

According to the results in Table 4.1, in a network setup using the MyriaNed nodes, slotted ALOHA performs much better. Thus, we used slotted ALOHA for accessing the MCS slots. Considering the number of clusters in our experiments and their mobility behavior, we set the length of the ALOHA super slots to $N_{AL} = 2$. This value is taken according to the calculations in Table 4.1. In these calculations, we used $\gamma = 2$ as the number of mobile clusters in each other's interference range. Fig. 4.9 shows the real distribution of the value of γ during our experiments. The vertical axis is the percentage of rounds that static nodes detect γ mobile clusters in their neighborhood. The graph presents the average over all static nodes. In around 75% of rounds, on average, no mobile cluster is in the range of the static nodes. This result is conform the power saving (70.5%) that is obtained by performing listening scheduling (as explained later). Only in less than 4% of rounds, multiple clusters are together (and may cause collisions) from which the majority is for two-cluster gatherings. It follows that considering $\gamma = 2$ is a proper design choice. Note that in the cases that a static node is in the range of no cluster or only one mobile cluster ($\gamma < 2$), no collision happens. Thus those cases are not of our interest in the evaluation of the protocol.

Fig. 4.9 also shows the distribution of γ in two nodes as examples of nodes that considerably deviate from the average because of their special locations. Node s_{26} is in the office of cluster C^1 and so in 90% of rounds senses one mobile cluster around. Node s_9 is located in the coffee corner where it is likely that several volunteers gather for some period of time. Thus the number of rounds that multiple clusters are in its neighborhood is much more than average. Later in this section we see that these nodes perform the most listening to the MCS section compared to the other static nodes.

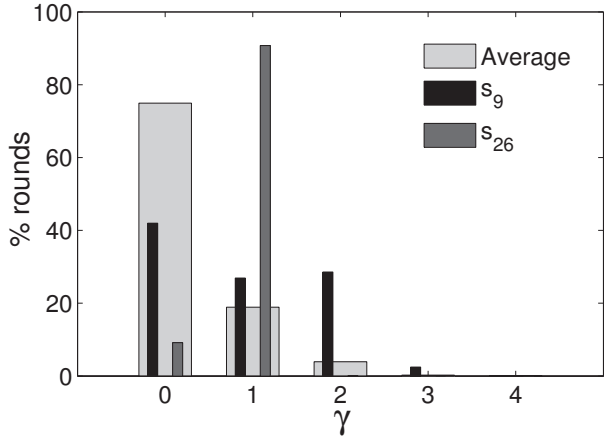


Figure 4.9: The distribution of the number of mobile cluster in each other's interference range (γ) in the experiments. The graph shows the average over all static nodes as well as the distributions in nodes s_9 (coffee corner) and s_{26} (C^1 office). For instance, in around 73% of rounds, the static nodes do not see any cluster ($\gamma = 0$) in their neighborhood, on average.

Table 4.2: Average values of some metrics over the whole experiments.

Metric	*static nodes*	s_{24}	C^1	C^2	C^3	C^4
1-hop PRR %	-	-	99	98	96	94
DDR %	86	77	71	94	88	88
Latency (second)	27	37	46	14	20	20

Packet delivery performance: To investigate the performance of the protocol, we consider various application-level metrics and detailed low-level metrics. The important issue here is that we should properly differentiate the effects caused by the routing protocol and the network deployment (e.g., network coverage and congestion) from the MCMAC performance. Table 4.2 presents the achieved 1-hop Packet Reception Ratio (PRR) over the whole experiment. The values are the average over the four nodes within each cluster. 1-hop PRR for a mobile cluster node is defined as the percentage of the packets transmitted by a mobile cluster node that have been directly received by *at least one* static node in the neighborhood. This is the main objective of the MCMAC protocol, to efficiently deliver mobile cluster node data to the static network. This metric is independent of the influences of the routing mechanism used in the static network because it only considers whether the packet enters the static part of the network. However, when a certain packet, sent by a mobile cluster node, is not received by any neighboring static node in the network, this may be caused by several issues, some outside the scope of MCMAC. From the MCMAC side, it may happen due to a wrong listening schedule of

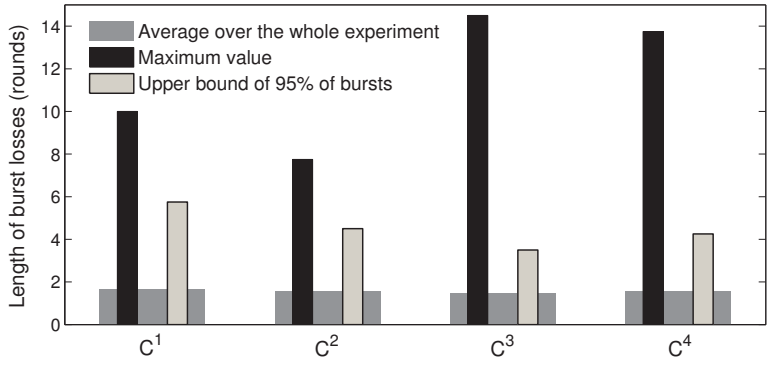


Figure 4.10: The length of packet loss bursts ($\mathcal{BL}^t(\eta)$) averaged over all nodes in each mobile clusters and over the whole experiment time. The graph also shows the maximum length of bursts and the least upper bound on the length in 95% of the bursts.

the static nodes around the mobile cluster, or because of a collision when several clusters are in each other's range. On the other hand, it may also happen due to other network circumstances such as link variations and interferences, or because the mobile cluster location is outside the coverage of the static network. However, the achieved 1-hop PRR values presented in Table 4.2 are quite promising.

The achieved end-to-end DDR ($\mathcal{R}^t(\eta)$) and latency (to the sink node) are also presented in Table 4.2 as high-level metrics. Although the routing mechanism and congestion in the static network have a very high influence on these performance metrics, they still provide an impression of the overall behavior of the protocol. For comparison, the table also shows the average values of these metrics over all static nodes as well as node s_{24}. This node is in particular considered because it is located near the office of cluster C^1. Thus we can compare the achieved end-to-end metrics for this node and cluster C^1 to check the behavior of the MCMAC protocol. The achieved metrics show that the DDR and latency of s_{24} are quite close to the values obtained for C^1 (considering the fact that C^1 is mostly one or two hops farther than s_{24} from the sink).

The lengths of bursts of losses ($\mathcal{BL}^t(\eta)$) is an important factor. It threatens the application performance when a cluster node cannot successfully deliver its packets to the static network for a long duration. Fig. 4.10 provides the average and maximum length of packet loss burst of the mobile clusters. Note that $N_{AL} = 2$; so if all four clusters gather together, the probability of successful transmission for a node in a round is $P_{ALOHA} = 1/8$. As Fig. 4.10 shows, the average lengths of packet losses for different clusters are very close (around 2 frames). This is because all clusters are in a similar environment taking the whole experiment duration into account. It is also worth pointing out that on average 89% of the losses are of one or two frames length and longer length burst losses are rare. Fig. 4.10 also shows the upper bound of the length of bursts in 95% of the burst losses during the experiments for each mobile cluster.

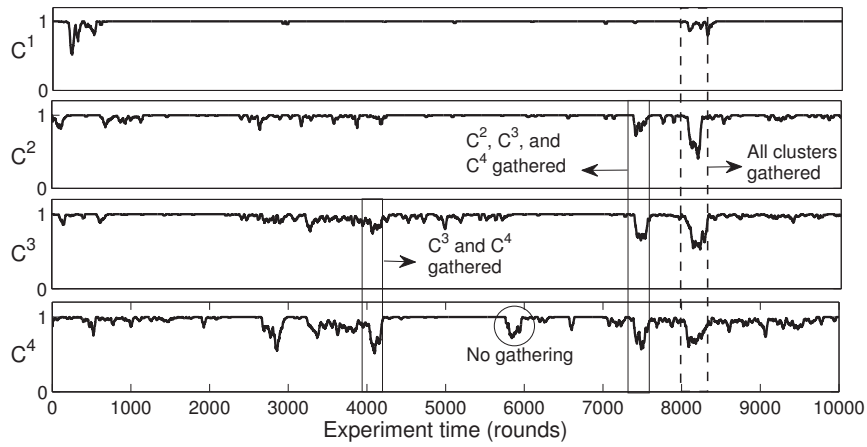


Figure 4.11: 1-hop PRR of the mobile clusters over a sliding window of size 20 rounds.

Fig. 4.11 shows the averaged 1-hop PRR over a sliding window of 20 rounds for the four mobile clusters. It gives a good view of the network behavior during the experiment. Several gathering durations of the mobile clusters are pointed out in the figure. The knowledge about the moments that gatherings take place is extracted from logged data. For instance, in the last gathering (rounds 8000-8300), all clusters are participating. Clusters C^3 and C^4 started the meeting. Then cluster C^2 joined. It also leaves the meeting earlier. Cluster C^1 is not fully participating, but joins intermittently. There are also some durations in which the 1-hop PRR of a cluster drops, but no cluster gathering takes place. This means that collisions are not the reason. One such instance is pointed out in Fig. 4.11 by a circle. Investigating this duration by checking logged data reveals that cluster nodes have not received any packets from the static nodes either. It shows that the cluster was out of the coverage of the static network during that time.

Listening scheduling behavior: We consider several low-level metrics to evaluate the performance of the listening optimization mechanism. Let E be the set of all TDMA rounds of an experiment, and let $\mathcal{L}_i \subseteq E$ be the set of rounds in which the static node s_i listens to the MCS part of the frame. $\mathcal{L}'_i = E - \mathcal{L}_i$ is then the set of all rounds in which the static node decides not to listen to the MCS part. Let $\mathcal{R}_i$ be the set of rounds that s_i is in the range of a mobile cluster and thus is able to receive packets from mobile cluster nodes. Note that during our experiments, we did not really turn off the receivers of static nodes the in MCS slots, in order to be able to compute $\mathcal{R}_i$. However, in the rounds that the node should not listen to the MCS part (according to the listening mechanism), the received packets are ignored for further processing. Similarly, $\mathcal{R}'_i$ are the rounds in which s_i is not in the range of any mobile cluster. These sets are estimated from the logged data for each node. Following are three metrics that we calculate for each node using these sets.

$$listening\ rate = \frac{|\mathcal{L}_i|}{|E|} \tag{4.15}$$

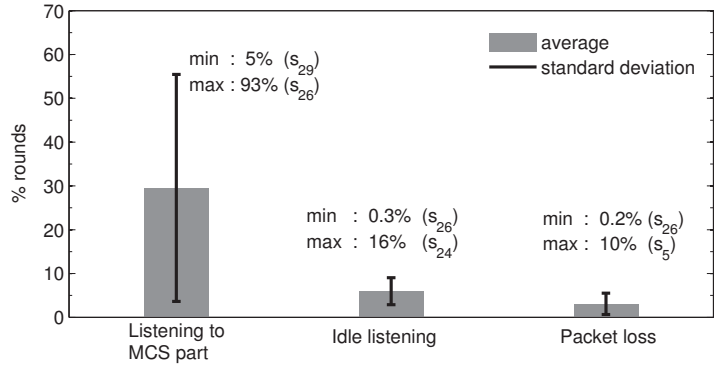
$$idle\ listening = \frac{|\mathcal{L}_i \cap \mathcal{R}'_i|}{|E|} \tag{4.16}$$

$$non\text{-}listening\ packet\ loss = \frac{|\mathcal{L}'_i \cap \mathcal{R}_i|}{|E|} \tag{4.17}$$

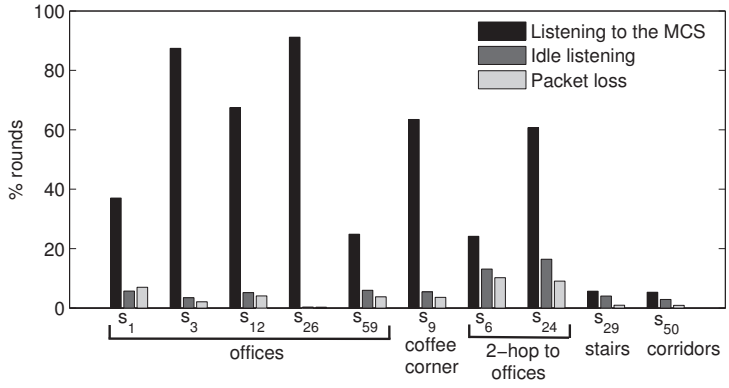
Listening rate is defined as the percentage of TDMA rounds in which the node listens to the MCS section of the frame. It reveals how our listening scheduling mechanism reduces listening activity of the static nodes for communicating with mobile clusters. However, in a part of the listening activities to the MCS section, the node receives no data packets. The *idle listening* metric computes the percentage of rounds that the node has been listening to the MCS section, but it has not received any data from any cluster node. This metrics shows to what extent the listening scheduling mechanism leads to idle listening avoidance. The last metric is *non-listening packet loss* that gives the percentage of rounds that the static node has not listened to the MCS section, but it would have received a packet from some cluster nodes if it had listened. Thus, this metric investigates the packet losses imposed by the listening scheduling mechanism.

Fig. 4.12(a) depicts the overall results regarding these metrics. We set the parameters as $T_{l,max}[2] = 3$, $T_{l,max}[i] = (i-2)\times 20$ for $i > 2$, $H = 24$, and $\alpha = 2.3$ to balance between idle listening and packet losses. These values are related to a Pareto configuration in Fig. 4.7(b). On average, static nodes have listened to the MCS part of the frame in 29.5% of the rounds during the experiments. This means a listening gain of 70.5% for the static nodes. Average idle listening and packet losses are 6% and 3%, respectively. However, the deviation per static node from the average listening rate is considerable. The achieved listening gain depends on the location of the static nodes. Nodes in or close to the offices of the volunteers (e.g., $s_1, s_2, s_3, s_{12}, s_{26}$) perform the highest rate of listening to the MCS part because there is mostly one cluster in their neighborhood. Moreover, these nodes have the least idle listening. Nodes located in 2-hop distance from the offices of the volunteers (e.g., $s_4, s_5, s_6, s_{15}, s_{24}$) have the most idle listening as they mostly anticipate a mobile cluster coming their way and so keep their listening interval low (that is why parameter $T_{l,max}[2]$ is so important). These nodes also have relatively high packet losses because, in many cases, the cluster in their 2-hop neighborhood suddenly comes closer. Nodes that are located in the corridors and stairs and are not close to volunteer offices or meeting rooms (e.g., s_{29}, s_{49}, s_{53}) perform the least listening activity.

Fig. 4.12(b) gives the obtained metrics for some selected static nodes in different locations. The volunteer wearing cluster C^4 has two working places (around nodes s_1 and s_{59}). The graph shows that cluster C^1 has spent most of the time in its office. That is why node s_{26} has the highest rate of listening to the MCS part among all static nodes.



(a) Average and distribution of the metrics over all static nodes.



(b) Metrics in some static nodes.

Figure 4.12: The obtained values of the evaluation metrics about performing listening scheduling mechanism in the experiments.

Comparison with M-LMAC behavior: There is no work in the literature that explicitly supports *cluster* mobility in WSNs. M-LMAC is a state-of-the-art protocol targeting *node* mobility; it is the best reference for comparison with MCMAC, to assess the impact of taking cluster mobility into account explicitly. Because the movements of mobile clusters in the experiments are based on daily activities of the volunteers, different experiments cannot have the exact same mobility patterns. Thus using different experiments to compare the performance of different protocols is inaccurate, considering the large statistical variation between similar experiments. We therefore compare the behavior of our protocol with M-LMAC performance by exploiting the logged data from the real experiments and simulating the M-LMAC protocol. This way, the mobility pattern, nodes neighborhood, and radio link status are the same as the real experiments done for MCMAC.

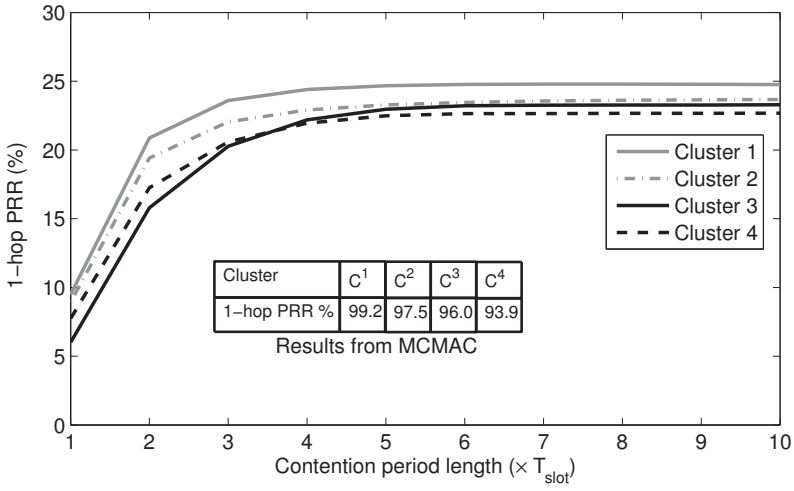


Figure 4.13: The overall 1-hop PRR for mobile clusters in the M-LMAC protocol using various lengths of the contention period. The table in the figure shows the 1-hop PRR achieved in the experiments running the MCMAC protocol (taken from Table 4.2).

In M-LMAC [98], communication with mobile nodes is initiated by the static nodes through transmitting an Announcement Message (AM). Mobile nodes that have data to transmit listen to the channel to find an AM, and then try to send their data by performing carrier sensing. Therefore, the number of opportunities that a mobile node has in each TDMA frame is equal to the number (say β) of static nodes in its communication range. We extract this number at each frame for mobile cluster nodes from the logged data. All mobile nodes in a neighborhood compete in these β contention-based access periods to send their data. The number of mobile clusters in a neighborhood at each frame is also extracted. Using this information, we calculate the probability of collision-free transmission at each frame for each mobile cluster node. We run the simulations for 10 different sizes of the contention period, as it is an important factor in M-LMAC. Each size is a multiple of the size of one TDMA slot.

Fig. 4.13 shows the 1-hop PRR for different clusters and various sizes of the contention period using M-LMAC. Comparing this result with the obtained results for MCMAC in Table 4.2 (reproduced inside Fig. 4.13) reveals the large gain in PRR obtained by our approach, due to the explicit consideration of *cluster* mobility, which avoids collisions between nodes in the cluster. Note that for the M-LMAC result, it is assumed that once a transmission without collision occurs, the packet will be received by the static nodes. This overestimates the PRR in M-LMAC as, in reality, other parameters like interference may further decrease the PRR. Despite that, the gap between the two protocols is significant.

Comparing only the PRR values is not enough, as there is a trade-off between PRR and energy consumption. To get an impression of the energy efficiency of the protocols,

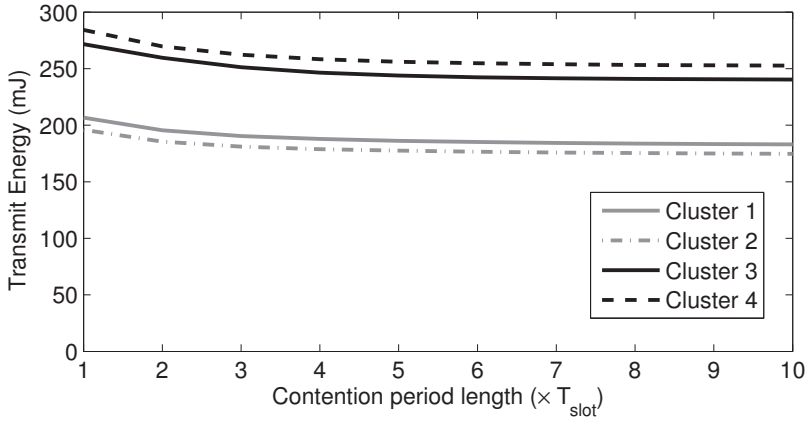


Figure 4.14: Transmit energy consumption of mobile cluster nodes during the whole experiment, performing the M-LMAC protocol for various lengths of the contention period. The values are the average over the 4 nodes in each cluster. The transmit energy consumption of the mobile cluster nodes in MCMAC during the experiment is 70mJ.

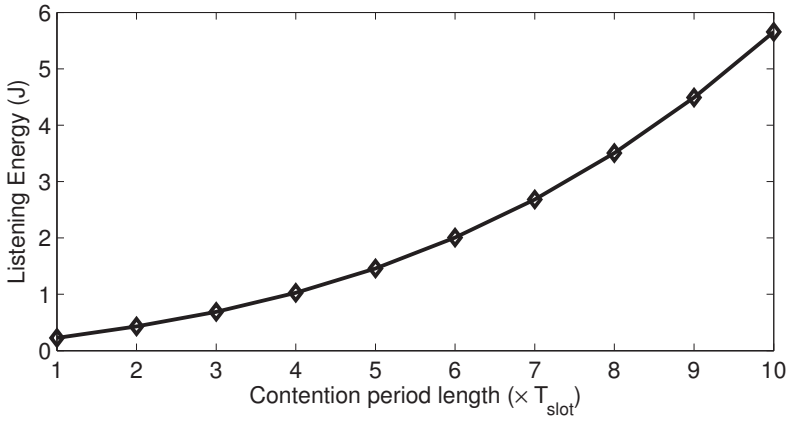


Figure 4.15: Average energy consumption of static nodes for listening to mobile nodes during the whole experiment, performing the M-LMAC protocol for various lengths of the contention period. The average listening energy consumption of static nodes in an MCMAC experiment is 541mJ.

Fig. 4.14 shows the transmit energy consumption of mobile cluster nodes during the whole experiment (10000 TDMA frames). In MCMAC, a mobile cluster node tries to transmit its packet only once in the MCS section, regardless of whether it is successful or not. In M-LMAC, the number of times that a mobile node tries to transmit its packet in each frame depends on the number of static nodes and the existence of other mobile nodes in its neighborhood. For each static node, all mobile nodes in the neighborhood compete to send their packet to the static node. At most one of these competing mobile nodes succeeds. The rest compete to send to the other static nodes in the neighborhood. Mobile nodes consume energy for carrier sensing too. As both M-LMAC and MCMAC (in the CSMA version) have this energy consumption, we do not include it in our comparison. The figure reveals a higher transmit energy consumption for mobile nodes in M-LMAC. Note that we have not considered the energy that the mobile nodes consume in M-LMAC to scan the channel for announcement messages. We assumed that (like in MCMAC), mobile nodes are synchronized with the TDMA framing and know the time slots for each static node.

Energy consumption of static nodes for receiving data from mobile cluster nodes is also very important. In M-LMAC, it heavily depends on the length of the contention period for each static node. When there is no mobile node around a static node, it listens for the whole duration of the contention period. This is also the case when a collision happens and thus the static node does not receive any packet. For each static node, we estimate the probability of collisions at each round using the number of mobile nodes in the neighborhood of the static node, extracted from the experiment's logged data. This is then used to estimate the number of slots in which a static node listens to receive data from mobile nodes in the M-LMAC protocol. Fig. 4.15 presents the listening energy consumption of static nodes in the whole experiment to receive mobile nodes' data. In the experiment using MCMAC, each static node listens to the whole MCS section in the frames that the listening scheduling mechanism asks for listening. For the whole experiment, listening energy consumption is 541mJ on average over all static nodes in the network. This energy consumption is close to that of M-LMAC with two slots as the length of its contention period. However, with this length of contention period, M-LMAC provides very poor 1-hop PRR in the setup of our experiments (Fig. 4.13).

In conclusion, MCMAC provides much better packet reception ratios at lower power consumption than M-LMAC (which, as one should remember, is not intended for cluster mobility). These results confirm that it is useful to provide specific support for cluster mobility in the MAC layer for WSN applications that exhibit this kind of mobility.

4.5.3 Simulations

We use the MiXiM [47] framework on top of OMNeT++ 4 [5] to further evaluate the MCMAC behavior by simulating several networks using this protocol. We considered three static network sizes, 6×6, 8×8, and 10×10. For each network, the static nodes are randomly distributed in a square area. To ensure a fairly even distribution across the area, we placed the nodes with a 10% variation around fixed grid points. While scaling the number of static nodes, the deployment area was scaled accordingly, such that the

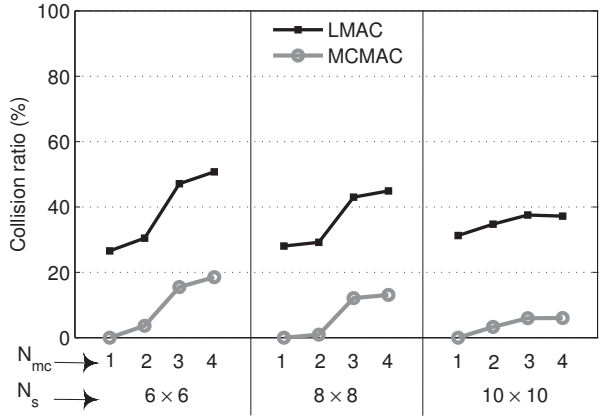
density of static nodes was equal for all networks. The static nodes in the corners were assumed to be the sink nodes. We performed simulations having various numbers of mobile clusters ($N_{mc} = 1, 2, 3, 4$). Each cluster contains five nodes and uses the MoBAN mobility model described in Chapter 3. Every node generates a new data item in each round (TDMA frame), which means a sampling period of one second. The cluster nodes use the CSMA scheme to access MCS slots. The length of the contention period is set to $T_{cp} = 1ms$. This provides a probability of 28% (Eqn. 4.4) for successful transmission for mobile cluster nodes when two mobile clusters are in each other's range. The presented results in this section are the average over several runs for each network setup.

Comparing the performance of M-LMAC and MCMAC in the experiments reported in the previous subsection, we learned that specific cluster mobility support is beneficial for certain applications. A qualitative reasoning suggests that LMAC may even perform better than M-LMAC in the networks that we are considering. In LMAC, nodes within a cluster will occupy different transmit slots (according to a unique slot assignment in the 2-hop neighborhood) and do not collide with each other. We checked this hypothesis for one of our network setups by running simulations using both LMAC and M-LMAC. The percentages of transmissions by mobile cluster nodes that failed due to collisions were 25.7% and 31.6% for LMAC and M-LMAC respectively. Because of that, in the simulations of this subsection, we focus on a pure LMAC scheme for comparison. For each network size we tried the LMAC scheme, our protocol without listening scheduling (MCMAC$^-$), and the full version of the MCMAC protocol. Simulations without listening scheduling enable us to investigate the impact of this mechanism on the performance of the protocol.

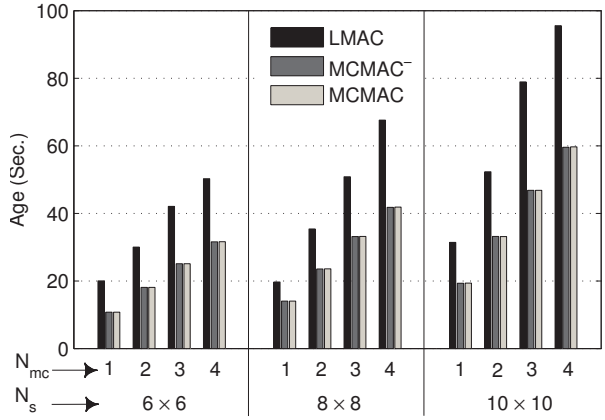
In contrast to the experiments using wireless nodes, the collisions can be counted during simulations in each node. This is done by checking if multiple packets have arrived in a single slot from different nodes. If so, all packets are ignored for further processing since a collision has happened. Fig. 4.16(a) shows the ratio of collisions detected by the mobile cluster nodes for all combinations of network sizes and cluster numbers. In general, the LMAC principle should provide a contention-free communication. But mobile cluster nodes may collide with static nodes due to their movement. For instance, in the setup with small network size and four mobile clusters using LMAC, the average collision ratio is 45%. There is also a chance of collisions in our MCMAC protocol when there are multiple clusters in a neighborhood, using the shared MCS slots for their data transmission.

We also consider the overall age of data items in the sink node to further evaluate the protocol. Fig. 4.16(b) shows the average age of the data items from cluster nodes. The first observation here is that the average age for cluster nodes using LMAC is 56% higher, on average, than the achieved average age in the MCMAC protocol. The main reason for such higher age is the collisions that happen using LMAC.

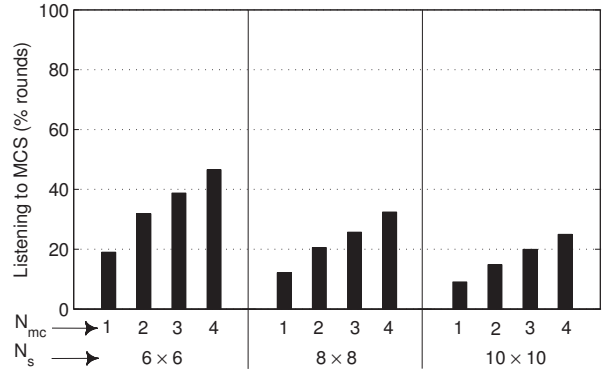
The second observation in Fig. 4.16(b) is that our listening scheduling mechanism does not exacerbate the age. The obtained values of the metrics are almost the same as in MCMAC$^-$ (the protocol without listening scheduling). Moreover, from the power consumption point of view, the saving is considerable. The average percentage of rounds with listening to the MCS part over all static nodes is shown in Fig. 4.16(c) for different setups. The parameters of the listening scheduling are set as $T_{l,max}[2] = 11$, $H = 15$, and



(a) Collision ratio (percentage of the received packets that are detected as collision).



(b) The average age of data items in the sink nodes running different protocols.



(c) The average percentage of rounds in which static nodes listen to the MCS part in the MCMAC protocol.

Figure 4.16: Simulation results for different network setups and protocols.

$\alpha = 1.25$ for all setups. This configuration is selected by trying several configurations and observing the behavior of the protocol in the simulation setup. The same configuration is set for all nodes. Static nodes have listened to the MCS part in 30% of rounds on average over all simulated networks. This means that the listening scheduling mechanism decreased 70% of listening to MCS slots, without causing QoS degradation to the network.

4.6 Discussion: Protocol Generalization and Future Work

So far in this chapter, we assumed only one MCS part in each TDMA frame. It is however possible to adapt the protocol for different circumstances by setting the protocol parameters according to Eqn. 4.4 and Eqn. 4.5 for the CSMA and slotted ALOHA versions, respectively. The density of the mobile clusters in the deployment area, cluster gathering tendency, and data delivery constraints in the application scenario are taken into consideration. To maintain a certain success probability of transmission for mobile cluster nodes, we may increase the length of the contention period (T_{cp}) in CSMA[2] or the number of slots per super slot (N_{AL}) in slotted ALOHA. However, this creates a trade-off between transmission success rate of the mobile cluster nodes and the energy consumption of the static nodes. Increasing these parameters is costly for static nodes as they listen to the whole MCS part in each time frame. On the other hand, not all mobile clusters are always together. Some parts of the network deployment may also have sparser existence of the clusters in some periods of time. A long MCS part causes an unnecessary high overhead for such static nodes.

Our suggestion here is to have multiple MCS sections in each frame for crowded networks (with many mobile clusters). This way, more energy saving can be achieved because the static nodes can then optimize their listening activities separately for different MCS sections, and each MCS section would be shorter. Let's assume that, in an application, we use $N_{sections}$ different MCS sections. Mobile clusters are grouped to use specific MCS sections. Different grouping approaches may be applied here. In one approach, an estimation of the number of mobile clusters that are together may be used. The clusters that are more likely to gather are assigned to different MCS sections to decrease the chance of interference. A second approach is to classify clusters based on their size. The size of the assigned MCS for a group is set as the maximum cluster size in that group. Clusters with similar or close sizes are then put into one group. Note that different MCS sections do not need to have the same length. The last approach can be based on QoS requirements for different clusters. The idea is to have smaller groups for clusters with stringent data delivery constraints. By this we decrease the chance of collisions for the nodes in those clusters. This approach can also be applied together with the first approach. These are the design time decisions about the number of MCS sections and the length of each one, and the approach for grouping clusters.

With multiple MCS sections, every static node needs to estimate and propagate $N_{sections}$ different hop-distances for scheduling its listening to the different MCS parts.

[2]For CSMA, the success probability has a limit as only one successful transmission is possible in each MCS slot

Thus d_i will be the estimated hop-distance to the nearest cluster that uses the i^{th} MCS part. A larger $N_{sections}$ obviously imposes higher communication and computation overhead on the static nodes to estimate corresponding d values. Taking the computation limits of typical nodes in WSNs into account, computation complexity is a considerable issue. In this sense, the number of MCS sections cannot be freely increased and a proper decision should be made. We consider exploring the trade-offs made by such an extension and its performance in dense application scenarios as a future work. With multiple MCS sections, every static node needs to estimate and propagate $N_{sections}$ different hop-distances for scheduling its listening to the different MCS sections. Thus d_i will be the estimated hop-distance to the nearest cluster that uses the i^{th} MCS part. A larger $N_{sections}$ obviously imposes higher communication and computation overhead on the static nodes to estimate the corresponding d values. Taking the computation limits of typical nodes in WSNs into account, computational complexity is a considerable issue. Moreover, in the worst-case, static nodes spend much energy for listening to different MCS sections. Therefore, the number of MCS sections cannot be freely increased and a proper decision should be made. We consider exploring the trade-offs made by such an extension and its performance in dense application scenarios as a future work.

4.7 Summary

This chapter presents a MAC protocol for supporting cluster mobility for TDMA-based protocols in WSNs. A network architecture consisting of static nodes and several mobile clusters is envisioned. The mechanism is specifically useful for the network architecture of health monitoring application considered in this thesis. The protocol dedicates a separate part in the TDMA frame to be used by sensor nodes in the mobile clusters (WBANs). A scheduling mechanism is exploited by static nodes for collision-free communication while mobile cluster nodes access the shared part using a contention-based mechanism. CSMA and slotted ALOHA are considered as two different paradigms for accessing the shared slots by the mobile cluster nodes. Guidelines for selecting the proper approach for a certain deployment and setup are provided. The behavior of the protocol is observed by performing extensive simulations with various network sizes and several real-world large-scale experiments using wireless nodes. The results show that this mechanism performs better in delivering data items from mobile cluster nodes to the static network and causes less collisions in comparison with similar protocols without specific cluster mobility support.

Idle listening to mobile nodes is a source of energy wastage in WSNs. A listening scheduling mechanism is proposed in this chapter to avoid idle listening to the mobile clusters when no cluster is in the neighborhood of the static nodes. This is done by estimating the hop-distance to the clusters and scheduling the listening activities accordingly. The experimental results show that this mechanism reduces listening to the mobile clusters around 70%, on average in the performed experiments and simulations. Moreover, such listening avoidance does not have a negative influence on the performance of the network.

Chapter 5

Data Dissemination in Heterogeneous Networks

5.1 Overview

All data samples generated by the static sensor nodes as well as the body sensor nodes (mobile cluster nodes) are supposed to be delivered to the sink nodes in the network. The MAC layer provides a platform for wireless sensor nodes to efficiently communicate and exchange data using their physical layer wireless radio facilities. On top of the MAC layer, a data dissemination or routing protocol provides end-to-end data delivery from sensor nodes to the sink nodes in one or more hops. Considering the short radio range of sensor nodes in WSNs, multi-hop data forwarding is very common. There are several approaches for designing data dissemination protocols; an appropriate approach depends on the network topology and application requirements. In general, robustness against node failures and node mobility, data delivery ratio and end-to-end latency, and overhead in terms of communication, computation, and memory footprint are important metrics in designing a data propagation mechanism for a WSN.

Heterogeneity is a common specification of distributed systems like WSNs. In health applications, in particular, there is a considerable heterogeneity in the network in terms of the type of the sensor nodes, surrounding environment, and QoS requirements. The spatial and temporal diversity in requirements and environment should be considered while designing communication protocols. Data routing and information dissemination protocols specifically should take this into account to meet the different QoS requirements. Data routing without attention to the heterogeneity may lead to very poor services for important information and an unnecessarily good service for information of lower importance.

Regardless of the type of routing protocol, a relaying node on a multi-hop routing path may have several data items waiting to be forwarded at any time. On the other hand,

the node may have a limitation in the amount of data that it can transmit in a given time duration. The limitation can be caused by the lower layer constraints like Medium Access Control (MAC) schedules or energy consumption limitations. This situation may happen more often for highly congested nodes like nodes closer to the sink nodes. So at any time, the node has to select a subset of the data items in its queue to transmit, and postpone the rest for transmission in the future.

In this chapter, we first present a dynamic priority assignment strategy for data dissemination aiming to support the heterogeneity in the network. QoS differentiation is the main objective. However, the proposed mechanism is able to provide a fair data delivery for different nodes over the network notwithstanding differences in their distance to the sink and mobility. Nodes on the data dissemination path toward the destination (sink) calculate priorities for the existing data items in their queue according to the relative requirements and the history of each data item. QoS requirements are defined individually for each data sample at the time of initiating the data item at the source node. This way, the QoS requirements are not labeled to the sensor nodes and so it can be changed over time allowing to handle temporal variations. We present and evaluate our dynamic data prioritization mechanism on top of a gossip-based data dissemination for the health monitoring application scenario. However, the scope of the proposed service is not limited to this specific routing mechanism and application scenario and can be used for many heterogeneous networks. The data prioritization mechanism was previously published in [58] and this chapter is mainly based on that.

The proposed data prioritization mechanism is evaluated using OMNeT++ simulations as well as real-world experiments. Various network setups with different QoS requirements and data sampling specifications are simulated to evaluate the mechanism in different scenarios. The protocol is also implemented on real wireless nodes (MyriaNed nodes) and several large-scale experiments with the same setup as that of the experiments in Chapter 4 have been performed. The aim is to observe the behavior of the prioritization mechanism and investigate its performance in QoS differentiation and to compare it with other approaches.

The next section discusses different sources of heterogeneity and possible kinds of diversity in typical WSNs. Health applications are taken into account as the main objective of this thesis. Related work for priority-based data dissemination (routing) in WSNs are reviewed in Sec. 5.3. The proposed dynamic priority-based routing mechanism and priority assignment strategy are presented in Sec. 5.4. Sec. 5.5 first describes the data dissemination protocol in which we integrate the proposed data prioritization mechanism. The section then presents the achieved performance evaluation results.

5.2 Heterogeneity in WSNs

There are many effects that make a WSN deployment heterogeneous. This diversity can be spatial or temporal. Spatial heterogeneity comes from diversity in circumstances in different locations of the network. The network characteristics vary over time, which leads to temporal heterogeneity. Here we review different possible sources of such heterogeneity in WSNs, in particular in our intended health applications.

5.2.1 Spatial Heterogeneity

Environmental heterogeneity: Sensor nodes in a WSN are deployed based on their functionality to sense specific parameters or to support enough connectivity for the network and prevent network disconnection. Due to nonuniform deployment, the nodes do not have similar neighborhoods. Nodes have different hop-distances to the sink nodes. A node may be able to send its data samples directly (one-hop) to the sink whereas data samples from some other node may need to traverse several hops to reach a sink. Accordingly, data traffic load depends on the location of the node and varies over the network. The network density is not uniform in different locations over the network coverage area. Moreover, quality of wireless links is not the same in different parts of the network area due to typical diversity in obstacles around the nodes, interference levels, directions of the nodes' antennas, and other environmental circumstances.

Different specifications of sensor nodes: In a deployment, we may use several kinds of wireless motes with different hardware features. Varying radio transmission power levels of different sensor nodes leads to different radio ranges. The type of energy source of the node (battery, energy scavenging, and so on) and its capacity, capabilities of the embedded processor for computations, and memory capacity are specifications of the sensor nodes that may vary from one node to another in a deployment.

Heterogeneity in sensing: Different sensors in a WSN may sense different parameters. The sampling specification depends on the nature of the signal being sensed and the application requirements for sampling precision and temporal resolution. In particular, the sampling rate of sensors may be different. This is a very important factor that should be considered to provide required QoS for data samples from a specific sensor nodes.

Heterogeneity in QoS requirements: In many applications of WSNs, there is a large variety between different sensor nodes in the network in terms of Quality-of-Service (QoS) requirements. Based on the specifications of the signal being sampled, the application may specify different latency constraints $\mathcal{LC}(\eta)$ and reliability constraints $\mathcal{RC}(\eta)$ in delivering data samples generated by different nodes η to the sink nodes. These QoS requirements specify the importance of different data samples. This kind of heterogeneity is specifically the objective of this chapter to be supported by the data dissemination protocol.

5.2.2 Temporal Variations

Environmental parameters and channel characteristics, neighborhood, sampling rate, and QoS requirements in a WSN deployment and its application are prone to changes over time. Mobility of nodes is an important factor that changes the neighborhood of the mobile nodes and the nodes around them. Mobility sometimes can substantially change the network topology and node density in some part of the network. The QoS requirements of a sensor node can also change over time according to the application scenario. As an example, consider a data propagation mechanism in which the requirements may change according to the value of the data samples. Multi-scenario applications is another example in which the behavior of the network, sensing specifications, and QoS requirements change over time based on the active scenario. For instance, in an ambient intelligence application, different scenarios might be used during day time and night time [90].

5.2.3 Heterogeneity in WSNs for Health Applications

Pervasive health monitoring applications show a considerable heterogeneity, especially in the QoS requirements and sensing specifications. In such applications, body sensor nodes sense and transmit biological signals of patients. In addition, the static sensor network is used to monitor the ambient parameters and to deliver both body and ambient information to the sink nodes. Besides the environmental heterogeneity in the network, there are differences between the static nodes and body sensor nodes in terms of the hardware features, sensing specifications, and the importance of the data samples. Data items from body sensor nodes are often of higher importance than ambient information. It means applications may have tighter latency and data delivery ratio constraints for body data than the ambient information. Loosing important ECG data or detected falls or very late delivery of such information may lead to severe danger for the patient. The sampling rates of ambient signals and body signals also deviate a lot from each other. Sampling the temperature of the living environment every 15 minutes is quite enough in a health monitoring application, whereas for some body signals such as ECG or EEG, multiple samples are taken per second. Such sampling differences also exist within a WBAN between different bio-sensors.

Note that in such applications, the body sensor nodes have high mobility (human body movements). Movement of the WBANs makes the hop-distance of the WBAN to the sinks vary over time. This changes the service for data samples from these sensor nodes. However, regardless of the location of the WBAN with respect to the sink nodes, the required QoS is expected to be continuously satisfied by the network.

Based on the health status of the patient, the QoS requirements may change over time. In a health application scenario, transmitting heart beat rate data once every few seconds may be enough for observing the health status of the patient when he/she is in a normal situation. However, in an emergency situation such as a heart attack, the whole ECG data may be necessary for intensive care of the patient. This situation may be detected by analyzing the EEG data by the body sensor nodes. Moreover, it is very likely that tighter latency and data delivery requirements are expected in emergency cases.

5.3 Related Work

Data prioritization has been used for data routing in several protocols for WSNs. In [53], a priority-based routing path selection mechanism is exploited for a proposed multi-path routing protocol (PRIMP), which is based on the directed diffusion [37] mechanism. During route discovery from a specific source towards a sink node, each discovered path is given a priority tag based on the length of the path in hops and the remaining energy source of nodes on that particular routing path. The source node then uses the priority tags of all paths to select the shortest and most robust route for data dissemination. The fundamental difference between this approach and our mechanism is that the PRIMP protocol does not prioritize the individual data samples. Instead, it tries to prioritize the available paths from a source node to the sink. Therefore all data items are treated in the same way without being prioritized.

The Priority-based Dynamic Adaptive Routing (PDAR) protocol as proposed in [21] aims to balance the energy consumption while providing better service for significant information. The protocol is based on an earlier routing protocol for multi-hop wireless ad hoc networks called Dynamic Source Routing (DSR) [40] with the emphasis on congestion prediction and priority scheduling for data routing. Data packets are categorized into two classes of vital and common packets. Accordingly, every node on the routing path maintains two separate data queues, each dedicated to a certain class of packets. The packets in the higher-priority queue (vital packets) are always sent before packets in the lower-priority queue (common packets). Compared to our data prioritization mechanism, in PDAR, a source node assigns a priority class (vital or common) to its data sample. In this way, the data item has a fixed priority class on the whole routing path regardless of the path length toward the destination. Therefore, data items from the source nodes far from the sink receive a poor service compared to the nodes in the sink's neighborhood. In our mechanism, the priority values are dynamically computed in each node on the routing path considering the data item trajectory and its QoS requirements. Moreover, PDAR uses a First-In-First-Out (FIFO) scheme to forward data samples in each priority class. In the case that there are multiple data items in the vital class, for instance, they are forwarded using the order of their arrival. Again, the time duration that the item has been on the path is not taken into account. Instead of making priority classes, we calculate a numeric priority value for all data items waiting to be forwarded and then select the items with the highest priority. The experiments presented in this chapter, show that a FIFO-based item selection is not able to provide appropriate services to different data samples with heterogeneous QoS requirements.

In [46], the Priority-based Hybrid Routing (PHR) mechanism is proposed in which data samples are classified by the source node into primary or secondary priority classes. This classification is done by observing the distinctiveness of the new sampled data in relation to the past data samples. A specific process using Dixon's Test [23] is performed on the periodically sampled data items to estimate the similarity of the present data with the past data. A data item is tagged as primary, if it shows considerable change with respect to previously sampled data, for example, when an abrupt change is observed in

the data stream. However, in PHR, priority classes are not used to order data items for forwarding. Instead, PHR uses different data dissemination schemes for items in different priority classes. A multi-path diffusion-based mechanism is used for forwarding the packets in the primary priority class to provide a more reliable and faster data delivery. A single-path routing mechanism based on the known Ad-hoc On-demand Distance Vector (AODV) [68] approach, that is prone to data loss, is exploited for items in the secondary class. This protocol is useful for application scenarios in which detecting changes in the parameters being sensed is very important. However, it does not distinguish data items from different sensors with different requirements. Like in the PDAR and PRIMP protocols, the assigned priority classes are fixed during the dissemination of the data item along the routing path. Our mechanism is also able to provide prioritization based on the changes in the sampled data stream by setting tighter QoS requirements for the data samples that show sudden changes with respect to the previous data items.

Each work mentioned so far has a specific criterion for assigning the priorities to data items and a particular means for providing proper services according to the priorities. However, none of them uses a dynamic priority calculation taking the status of the data item on the routing path and its QoS requirements into consideration. In this chapter, we propose a mechanism to *dynamically assign priorities to data items waiting to be forwarded at any node regardless of the type of routing structure with a focus on considering dynamic heterogeneity in the network.* The mechanism aims to provide differentiated services for data items according to their QoS requirements.

We also consider scenarios in which the requirements change over time. To provide such a flexibility, instead of attaching priority values to the data packets, relative QoS requirements are attached to the individual data items by the source nodes. Then the priorities are calculated at each relay node on the routing path taking the attached QoS requirements, and the history of the data item into account. This way a source node can change the requirements for its data items at any time. In addition, as the history of the data item (for instance the time it spent on the path) is taken into account, dynamic priority calculation provides appropriate services for nodes farther away from the sink node. This is specifically interesting for mobile nodes for which the hop-distance to the sink node varies over time.

5.4 Dynamic Data Prioritization

In this section, we present our approach for data routing taking the individual QoS requirements of the data items into consideration. First, the overall approach for data item selection is presented. We then define the QoS requirements that we consider and explain how these requirements are announced by the sensing nodes. The strategy used for priority assignment is then presented.

5.4.1 Priority-based Data Forwarding

To have a better understanding of the proposed prioritization method, we recall the main goals of exploiting such a mechanism. The first goal is to disseminate or route data in a dynamic and heterogeneous WSN in order to distribute the network capacity according to the relative QoS requirements for data items. Note that one can state QoS requirements that cannot be met considering the situation in the deployed network. Consider the following example as a very simplified case. Satisfying a latency requirement of $\mathcal{LC}(\eta) = 5$ seconds for a node η that is 7 hops away from the nearest sink is not feasible (assuming a transmission period of 1 second for the nodes in the network). In the best case in which no packet loss happens and there is no other traffic load in the network, it takes at least 7 seconds for a data sample from node η to reach the sink node. Our goal is that given a network deployment with heterogeneity in the requirements and a communication protocol stack, we extract useful information to detect relatively more important data items at any time to be forwarded earlier. The second goal is to provide the flexibility to have time-variant QoS requirements. The sensor nodes generating data items can then change their requirements based on the situation.

Assume that the static sensor node η has k data items waiting to be forwarded. It is able to transmit a limited number of items (λ items) in a given time duration. If $k \leq \lambda$, node η can forward all items. However, there may be congestion on the routing path. The data traffic depends of the position of the node and the network data load. In the case that $k > \lambda$, a selection mechanism must be applied.

A uniformly random selection strategy is used in [29]. In this strategy, a statistically equal chance is given to all data items waiting to be forwarded. Besides that, in [29], the nodes always send their own data items. Moreover, if a data item is going to be removed from the local memory because of lack of storage space, it will be transmitted hoping that some other nodes store it. However, this may not be the best option for the applications with different QoS requirements for different data items. Our approach is to assign priority values to the k data items waiting to be forwarded and then select λ items with the highest priority values. The priority values are assigned dynamically according to the QoS requirements of individual data items, and their history on routing paths so far.

5.4.2 QoS Requirements

Each source node has particular requirements for its sampled data. In general, we assume that $\overline{\mathbf{Q}} = (Q_1, \cdots, Q_d)$ is the vector of values of d different QoS requirements that are of interest in the running application and are specified for each data item. Suppose that the sensor node η initiates a new data item $\Gamma_\eta^{t_s}$ at time t_s. The node η decides about its QoS requirements and includes the specified vector $\overline{\mathbf{Q}}_\eta^{t_s}$ into the data item. For priority calculation by the nodes on the routing path, some application dependent parameters may also be needed to be attached to the data items. The sampling time of the data item at the source node (t_s) in our application scenario is an example.

The amount of overhead of adding QoS requirements $\overline{\mathbf{Q}}_\eta^{t_s}$ and other parameters to

the data items depends on the level of heterogeneity in the requirements in the network. Examples of the requirements are discussed later in this section. An appropriate quantization method is selected to represent this information with a limited number of bits, which does not need to be uniform. For instance, if we have four different levels for Q_1 in a scenario, it can be represented by just two bits in each data item. All nodes on the routing path are then able to compute the value of Q_1. Moreover, sometimes the required parameters may overlap with the existing data payloads in the routing mechanism. For instance, when we require the sampling time of each data item for priority calculation while it already exists in the payload of the data items in our application scenario.

The QoS metrics and their interpretation depend on the requirements of the running application and cannot be unique for all applications. To better illustrate the mechanism, we specifically consider reliability and latency requirements (defined in Chapter 2) as two common QoS metrics. Here we define the QoS requirements for each individual data item to also illustrate the support for time-varying QoS requirements. Thus, we consider $0 < \mathcal{RC}(\Gamma_\eta^{t_s}) \leq 1$ as the reliability requirement of a particular data item $\Gamma_\eta^{t_s}$, which is the probability that the data item should reach a sink node. Similarly, the latency requirement $\mathcal{LC}(\Gamma_\eta^{t_s})$ is a constraint on the latency of data item $\Gamma_\eta^{t_s}$. If this constraint cannot be met, we assume that the application needs to minimize the lateness $(\mathcal{L}(\Gamma_\eta^{t_s}) - \mathcal{LC}(\Gamma_\eta^{t_s}))$ of this data item as much as possible.

The node η includes $\overline{\mathbf{Q}}_\eta^{t_s} = (\mathcal{LC}(\Gamma_\eta^{t_s}), \mathcal{RC}(\Gamma_\eta^{t_s}))$ as the required latency and reliability into the data item that it samples at time t_s. We also include the sampling period $T_s(\eta)$ of node η. This parameter informs the relaying nodes on the path about the time of initiating the next data item by node η. This parameter is required for our data prioritization mechanism because in the base gossip-based mechanism, an older data item of node η will be overwritten by the received newer item even before transmission. This is because only the most recently generated data item of node η can be stored in the cache of relaying nodes on the routing path at any time. We use the sampling period to estimate the time that the data item may be overwritten by a newer data item from the same node.

5.4.3 Dynamic Priority Assignment

In this section, we present a method to calculate priority values for data items using their relative QoS requirements, the time that the item has already been on the routing path, and the time that the data item may be overwritten by the next item. Higher priority means higher chance of being selected to be forwarded by the current relaying node. Earlier transmission leads to lower end-to-end latency and a lower chance of being lost due to being overwritten. We first calculate a priority value with respect to each metric separately. Thus, we obtain a vector $\overline{\mathbf{P}} = (p_1, \cdots, p_d)$ called *partial priority* values for each data item in which value p_l is the calculated priority value related to the QoS requirement Q_l where $1 \leq l \leq d$. The partial priorities are calculated considering the type of the metric and its interpretation in the running application. Later in this section, we present a method for calculating partial priority values for latency and reliability. Finally, we have k partial priority vectors for k existing data items denoted by $\overline{\mathbf{P}}_m$ where

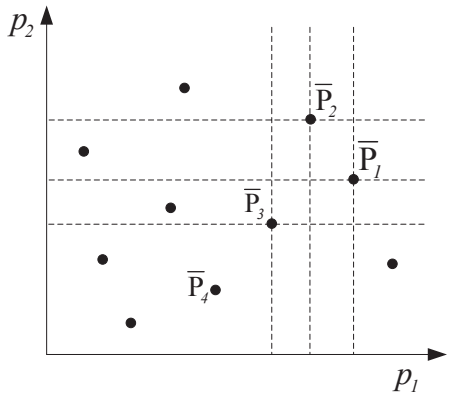


Figure 5.1: An example of the obtained partial priority vectors in a two-dimensional space for nine data items waiting to be forwarded.

$1 \leq m \leq k$. The method for calculating partial priority values for latency and reliability is also presented later in this section.

The next step is to extract a single priority value for each data item from its calculated partial priority vector as follows.

$$P_m = f(\overline{\mathbf{P}}_m) \ , \ 1 \leq m \leq k \tag{5.1}$$

The specific function can be chosen according to the desired behavior of the protocol or QoS. However, the function $f(.)$ should obviously be monotone by providing higher priority values for *Pareto dominating* [66] partial priority vectors. The partial priority vector $\overline{\mathbf{P}}_{m_1}$ dominates vector $\overline{\mathbf{P}}_{m_2}$ denoted $\overline{\mathbf{P}}_{m_1} \succeq \overline{\mathbf{P}}_{m_2}$, if none of the individual priority values in $\overline{\mathbf{P}}_{m_1}$ is less than the corresponding one in $\overline{\mathbf{P}}_{m_2}$. Monotonicity means that if $\overline{\mathbf{P}}_{m_1} \succeq \overline{\mathbf{P}}_{m_2}$ then $f(\overline{\mathbf{P}}_{m_1}) \geq f(\overline{\mathbf{P}}_{m_2})$ for the whole space of priority vectors.

Fig. 5.1 depicts an example of the obtained partial priority vectors with two QoS metrics. In this example, $\overline{\mathbf{P}}_3 \succeq \overline{\mathbf{P}}_4$ and so using the function $f(.)$, it is expected to come to priority values such that $P_3 \geq P_4$. Similarly, we should have $P_1 \geq P_3$ and $P_2 \geq P_3$. But the comparison between vectors $\overline{\mathbf{P}}_1$ and $\overline{\mathbf{P}}_2$ is not straightforward as none of them dominates the other one. The relative priority values in these cases should be calculated according to the criteria in the running application and the nature of the QoS metrics, for example as a weighted average of prioritizing criteria.

For the latency requirement of the data item in our application, we consider at time t the remaining time to expiration of the latency requirement, which is $L_\eta^{t_s} - (t - t_s)$. Less remaining time to the latency deadline inspires a higher priority for the data item. Nevertheless, there is a chance that the data item is not selected due to high congestion at the relaying node and the presence of other items of higher priority; eventually the deadline may expire. In such case, we may continue increasing the priority of the data item to give it a higher chance to be selected as soon as possible. We assume that the data

item still has value even if the latency deadline has expired. Thus we try to minimize the final latency of delivering the data item to the sink[1]. To calculate the partial priority of data item $\Gamma_\eta^{t_s}$, denoted by $p_1(\Gamma_\eta^{t_s})$, related to the latency requirement, we first compute an intermediate parameter, T_c, which expresses the remaining time for this data item to expiring the latency requirements, normalized with respect to the transmission period T_{tx}. Eqn. 5.2 computes T_c. This parameter T_c decreases to 1 when the latency has not yet expired (first case in Eqn. 5.2), if we assume that all nodes use the same transmission period and the sampling period of all nodes is a multiple of transmission period. The normalization with respect to T_{tx} leads a minimum value of 1 for T_c in the first case. Note that priority calculation is done periodically by the relaying node before each packet transmission. Thus the time parameter t in Eqn. 5.2 gets values that are multiples of the transmission period. After the latency constraint expires (second case in Eqn. 5.2, if it happened), T_c decreases hyperbolically to 0 to give a higher priority to the data item.

$$T_c(\Gamma_\eta^{t_s}) = \begin{cases} \dfrac{\mathcal{LC}(\Gamma_\eta^{t_s}) - (t - t_s)}{T_{tx}} & \mathcal{LC}(\Gamma_\eta^{t_s}) > t - t_s \\ \dfrac{\mathcal{LC}(\Gamma_\eta^{t_s})}{(t - t_s) \cdot T_{tx}} & otherwise \end{cases} \tag{5.2}$$

A lower value of parameter T_c means less time to violation of the latency requirements. Thus we take it as a notion of higher priority for transmission, hoping that the item gets higher chance for meeting its latency constraint. Therefore, the partial priority of data item $\Gamma_\eta^{t_s}$ regarding the latency requirement is set as follows.

$$p_1(\Gamma_\eta^{t_s}) = \frac{1}{T_c(\Gamma_\eta^{t_s})} \tag{5.3}$$

Similarly, for the prioritization on reliability, we consider the expected time for the data item of node η to be overwritten by the next data item of this node. When a more recently sampled data item from node η arrives, the current one is overwritten. Thus the probability that this item reaches the sink node reduces. Therefore, we aim to give a higher priority to the data items that we expect to be overwritten earlier. We first make an estimation of the arrival time of the next data item considering the current sampling period T_s of the source node η. Suppose that the data item $\Gamma_\eta^{t_s}$ arrives at the current relaying node at time $t_a(\Gamma_\eta^{t_s}, s)$. The time period that this item has been waiting in the current relaying node at time t is $t - t_a(\Gamma_\eta^{t_s}, s)$. Longer time of waiting means higher chance of the arrival of the next item before transmission of the current item, which also depends on the sampling period. We use $T_s(\eta) - (t - t_a(\Gamma_\eta^{t_s}, s))$ as an estimation of arrival of a newer data item. However, a newer data item may arrive earlier or later considering

[1] In a different application scenario, which we do not work out here, the fraction of data items that reach the sink within the latency constraint may be of higher importance. In such a scenario, once the latency constraint has expired for a data item, its priority should be reduced to improve the chances for other items that may still make their deadline.

differences in the dissemination process of the current data item $\Gamma_{\eta}^{t_s}$ and the successor items $\Gamma_{\eta}^{t_s+n.T_s}$ $(n \in \mathbf{N})$ on the routing path to the current node.

To calculate a partial priority value related to the reliability requirements for data item $\Gamma_{\eta}^{t_s}$, we first calculate a parameter $T_o(\Gamma_{\eta}^{t_s})$ that expresses the estimation of the expected time till $\Gamma_{\eta}^{t_s}$ being overwritten, normalized with respect to the transmission period T_{tx}. If a newer data item is not received after $T_s(\eta)$, T_o goes below one and it keeps decreasing. It means that we are going to give higher priority to this item as it might be closer to being overwritten. Eqn. 5.4 gives the calculation of the parameter T_o.

$$T_o(\Gamma_{\eta}^{t_s}) = \begin{cases} \dfrac{T_s(\eta) - (t - t_a(\Gamma_{\eta}^{t_s}, s))}{T_{tx}} & T_s(\eta) > t - t_a(\Gamma_{\eta}^{t_s}, s) \\ \dfrac{T_s(\eta)}{(t - t_a(\Gamma_{\eta}^{t_s}, s)) \cdot T_{tx}} & otherwise \end{cases} \tag{5.4}$$

Lower value of T_o inspires giving a higher priority to the data item for transmission to reduce the chance of being lost. Besides considering this time estimation, we consider the number of times that the data items from the a node η have been successively overwritten without being transmitted. Due to possible congestions, a data item may not get a chance to be forwarded before the arrival of the next data item from the same node. In this case, we aim to give higher priority to the new data item, hoping to reduce the length of the burst data losses and provide higher DDR for the source node. The partial priority related to the reliability requirements is then obtained by Eqn. 5.5.

$$p_2(\Gamma_{\eta}^{t_s}) = \frac{\mathcal{RC}(\Gamma_{\eta}^{t_s}) \times (1 + N_o^{\eta})}{T_o(\Gamma_{\eta}^{t_s})} \tag{5.5}$$

where $N_o^{\eta} \geq 0$ denotes the number of times that data items from node η have been successively overwritten without being transmitted. Once a data item from node η is selected for transmission, the value of N_o^{η} will be reset. The term $1 + N_o^{\eta}$ is used in Eqn. 5.5 to increase the priority for bigger N_o^{η}. Note that N_o^{η} is initially zero and we add one to it in the equation to avoid a value of zero for the partial priority when $N_o^{\eta} = 0$. After all, the reliability requirement for data item $\Gamma_{\eta}^{t_s}$ also plays a role. A higher value of $\mathcal{RC}(\Gamma_{\eta}^{t_s})$ leads to a higher partial priority value in Eqn. 5.5.

Both partial priority values calculated so far are related to time. Higher priority means that there is less time to expiration of the deadline if it has not been reached yet. We take the maximum of these two partial priority values for each data item to force the mechanism to select the items closer to their deadline either related to the latency or reliability. Therefore, the priority value for the data item $\Gamma_{\eta}^{t_s}$ is obtained as follows.

$$P(\Gamma_{\eta}^{t_s}) = \max\{p_1(\Gamma_{\eta}^{t_s}) \ , \ p_2(\Gamma_{\eta}^{t_s})\} \tag{5.6}$$

The maximum function has the monotonicity property. Each data item waiting to be forwarded in the current relaying node is given a priority value calculated by Eqn. 5.6. When the node wants to send a packet, it selects λ data items with the greatest priority values to include in the transmission packet.

5.5 Performance Evaluation

We investigate the performance of our prioritization mechanism for data dissemination in heterogeneous and time-varying networks through various computer simulations and experiments. Several simulation setups are also developed to show the ability of the mechanism for QoS differentiation and heterogeneity support. For real-world experiments, we used a similar setup as that of the experiments in Chapter 4 for performance evaluation of the MCMAC protocol.

5.5.1 Used Mechanism for Data Dissemination

In the health application that we are considering, there are two different data flow directions in the network. First, all sampled data from all the sensor nodes in the network should be gathered at the sink node. This is an all-to-one dissemination scheme. We assume only one sink node in the network for the experiments in this section. The other data flow direction is broadcasting information from the sink node to all sensor nodes. This information can be control commands, or information of different communication layers including the application itself. This is a one-to-all dissemination scheme. As a case study, we use the gossip-based data propagation mechanism presented in Chapter 1 as the base to establish both types of data flows for our health applications. The protocol is used on top of the MCMAC layer. A directed gossiping strategy is used to forward sensed information toward the sink node. On the other hand, a pure gossip-based data dissemination is exploited for sink data to be disseminated to all nodes in the network. This data dissemination is explained in the following.

Every static sensor node maintains a pool of data items that it has received and has to forward. In the experiments, we assume that every node has sufficient memory to store one data item (the newest received item) from each node in the network. Note that this is not a fundamental assumption for the proposed dynamic data prioritization mechanism. Sink data items are recognized by the source ID in the structure of the data item. To the aim of directed gossiping of sensed data items towards the sink nodes, an underlying minimum cost computation service is used [89]. In a simple case, the cost can be the hop-count to the sink node. Using such a service, static nodes know their *cost* to reach the sink and use that for data forwarding. The nodes also include their cost in their transmission packets so that the receiver node knows the status of its neighbors to make decisions about the arriving packets. Because of the high mobility of mobile clusters (WBANs) in the network, the cost of mobile body nodes to the sink will change frequently and is not reliable. Therefore, it is not calculated and not used for these nodes.

Every static node may receive several packets at each round, each including several data items. Algorithm 4 shows the process that a node performs upon receiving a packet. The data items *directly* received from a body sensor node as well as data items originated by a sink node are processed without considering the cost to the sink. For other received data items, if the cost of the current node to the sink is less than the cost of the direct transmitter of the packet, the item is considered for storing in the data pool. Otherwise, it means that the item is transmitted from a node closer to the sink and so the current node

Algorithm 4: Processing the received packet in the data dissemination mechanism.

```
Data:
pkt: Received packet
x.t_s: Sample time of the data item x
x.src: Source node of the data item x
TxNode: Direct transmitter of pkt
Cost: Cost to the sink (minimum hop-count)
ReceivePacket(pkt)
foreach RxItem ∈ pkt do
    CItem = RetrieveFromDataPool(RxItem.src);
    if IsSink(RxItem.src) ∨ IsMCNode(TxNode) then
        if RxItem.t_s > CItem.t_s then
            StoreInDataPool(RxItem);
        end
    else
        if (TxNode.Cost ≥ Cost) then
            if RxItem.t_s > CItem.t_s then
                StoreInDataPool(RxItem);
            end
        end
    end
end
```

will not participate in forwarding that item. Note that a received data item originating from node *src* will be stored in the pool only if it is newer than any existing one from node *src*. Besides this, the node inserts its own data into the pool whenever it samples new data.

The structure of a data packet in the communication protocol is shown in Fig. 2.3 in Chapter 2. At every round (TDMA frame), each node makes a packet by selecting λ data items from its pool and delivering them to the MAC layer to be transmitted in its dedicated transmit slot. The selection mechanism has a very strong impact on the network performance. The base dissemination scheme of MyriaNed nodes uses a uniformly random selection, except for its own items, which is always included in the transmission packet. We use the presented priority assignment strategy to assign a priority value to each data item existing in the pool and then select the λ items with the highest priority. In the case that there are several items with the same priority value and insufficient space to include all in the packet, we use a uniformly random selection.

5.5.2 Simulations

To investigate the behavior of the prioritization mechanism in different circumstances, we run extensive simulations with various network setups. Each simulation was run for 4000 rounds (TDMA frames). All results shown in this section are the average over all runs. The MoBAN mobility model, presented in Chapter 3, is used for moving the WBANs within the simulation area. The same mobility pattern has been used for all simulation runs. To the aim of comparison, we run simulations using our prioritization mechanism, uniformly random, and First-In First-Out (FIFO) item selection strategies.

Table 5.1: QoS requirements and sampling periods of sensor nodes in simulation Setup1 ($1 \leq i \leq 100$, $1 \leq k \leq 4$)

class	**nodes**	T_s	$\mathcal{LC}$ (*Latency req.*)	$\mathcal{RC}$ (*Reliability req.*)
Class1	s_i (static nodes)	20 *sec.*	70 *sec.*	0.5 (50%)
Class2	c_1^k , c_2^k	3 *sec.*	20 *sec.*	0.8 (80%)
Class3	c_3^k , c_4^k , c_5^k	10 *sec.*	40 *sec.*	0.7 (70%)

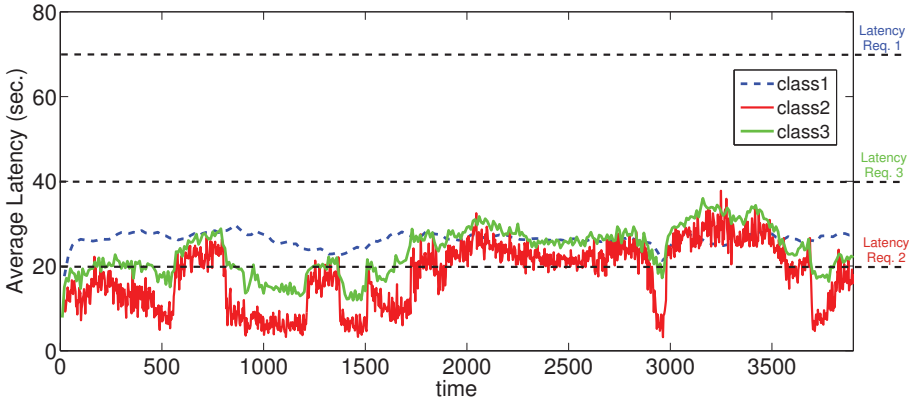
We simulate the prioritization mechanism in three different setups. In Setup1, there is heterogeneity in the sampling rates and QoS requirements between different nodes. However, these specifications remain fixed during the simulations. This setup aims to investigate the behavior of different item selection schemes when there is spatial heterogeneity in the network. In Setup2, the QoS requirements of the mobile cluster nodes change in some periods to observe how our prioritization mechanism reacts to temporal dynamics in the requirements. In the first two setups, a grid topology is used for the static network. Setup3 uses the deployment of a real COPD experiment for its static network deployment. There is a higher level of heterogeneity in this setup and there is also diversity in the requirements of the static nodes. In the following, these setups and results are discussed.

Simulation Setup1: Fixed QoS Requirements

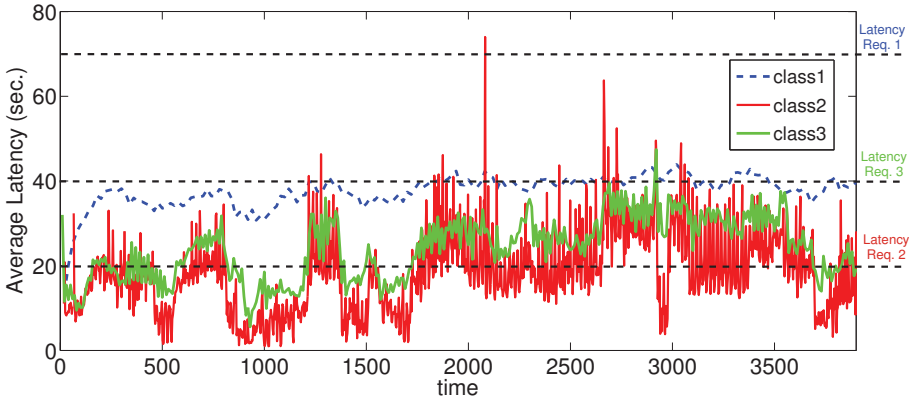
The first simulation setup aims to investigate the effect of using the data prioritization mechanism by comparing to runs without using priorities. We randomly distributed 100 static sensor nodes in a square area. The nodes are placed with a 10% variation around fixed grid points. The sink node is placed somewhere around the middle of the area. Four WBANs ($N_{mc} = 4$) are considered, each containing five sensor nodes.

The simulation employs three classes of QoS requirements that remained fixed for the whole simulation run. The first class includes all static nodes (ambient sensors) that are more delay tolerant with the lowest packet delivery requirements and the lowest rate of data generation. Mobile cluster nodes are categorized into two classes. Two nodes of each cluster (WBAN) have the highest sampling rates with tightest requirements while the other three require less service. Table 5.1 presents the values of sampling rates and QoS requirements we used.

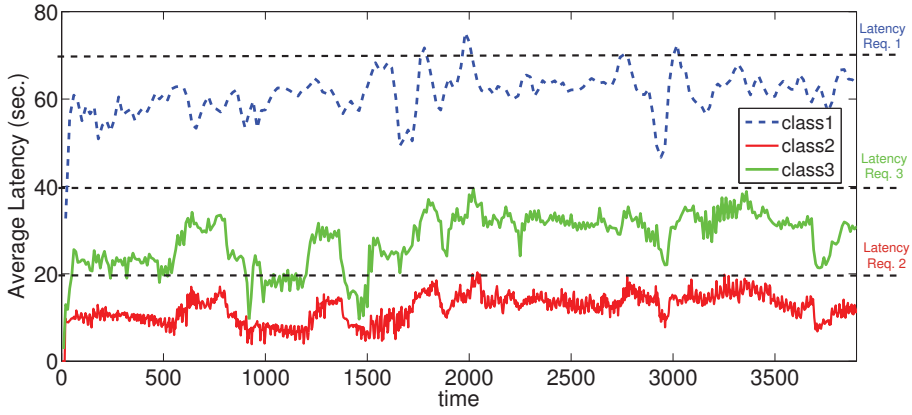
Fig. 5.2 depicts the latency of the arrived data items at the sink node for simulation Setup1. The horizontal axis is the time of production of data items. The latency values are average values over all nodes in each class. Horizontal dashed lines in the figure show latency constraints for different classes. Note that a part of the variations in the latency is caused by the movement of WBANs that are sometimes far away or sometimes close to the sink node. Table 5.2 presents the achieved DDR values, averaged over all nodes in each class, for the whole simulation time. It shows the achieved DDR together with the DDR requirements of each class to provide an easier analysis of the behavior of each method in sharing the network bandwidth. The table also shows the average latency violation in each class.



(a) Uniform random data selection



(b) FIFO-based data selection



(c) Priority-based data selection

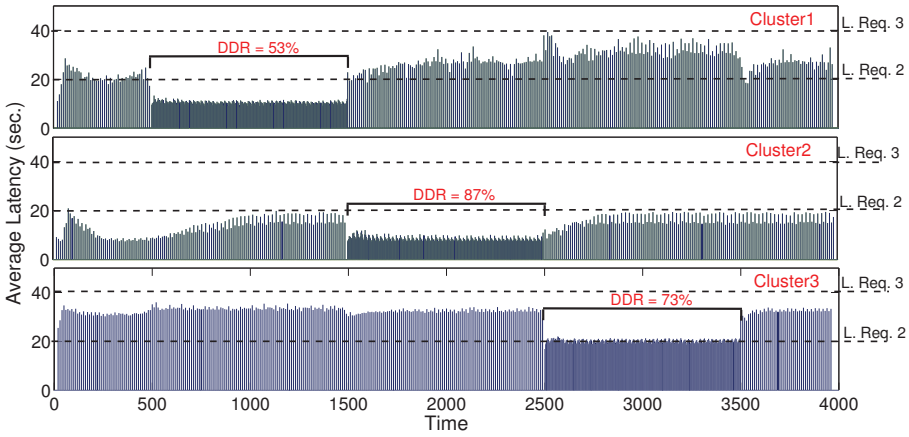
Figure 5.2: The average latency of received data items from all nodes in each class in simulation Setup1.

Table 5.2: Average DDR over all nodes in each class and the average latency violations using various methods in simulation Setup1.

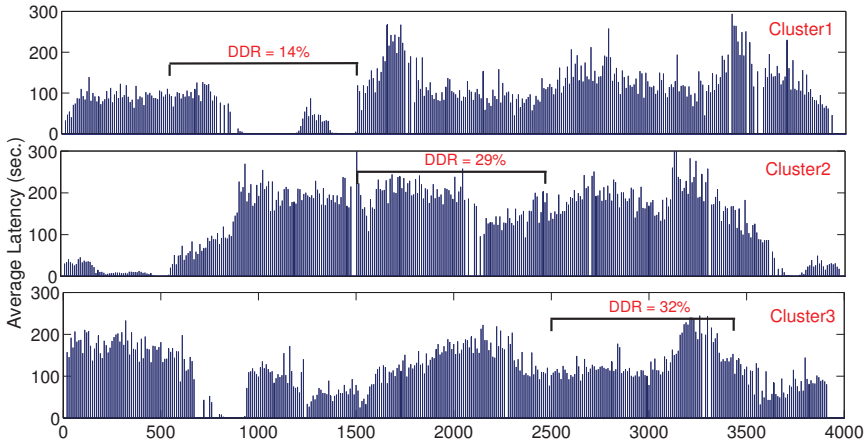
Class	$\mathcal{RC}$	*avg. DDR (%)*			*average latency violation (rounds)*		
		Random	*FIFO*	*Priority*	*Random*	*FIFO*	*Priority*
Class1	50	80	81	45	5	370	309
Class2	80	24	28	63	1935	2106	789
Class3	70	57	55	69	485	1053	660

Fig. 5.2(c) shows the results when our priority-based data selection is used. It is observable that the bandwidth has been shared according to the requirements of each sensor class. The mechanism has provided differentiated latency for different classes that correspond to their latency requirements. In contrast, the result obtained from the random item selection (Fig. 5.2(a)) shows that this item selection scheme treats all items from different classes in the same way. Table 5.2 says that the provided DDR for Class2 (highest demanding nodes) is much lower than the requirements for this class. Although the shown latency values are still close to the constraints, these values are only for the received data items. The majority of items have been lost (DDR=24%). But, good DDR (80%) and latency values have been achieved for Class1, which are not necessary at all. Since the sampling rate of nodes in Class1 is the lowest, the chance of being overwritten is low. That causes a good DDR for those data items. This kind of bandwidth sharing is what our priority-based item selection tries to avoid. Fig. 5.2(b) shows that a similar (even worse) result is obtained when a FIFO-based item selection strategy is applied. This strategy only takes the arrival time of data items into account for forwarding them. This way, the items with high demands may be delayed for forwarding less important items.

Table 5.2 also gives a metric that shows the amount of latency violation in each simulation. When the latency of an arrived data item to the sink node is violated, the difference between the achieved latency and the latency requirement for that data item is computed. We then add up all latency violation values for all received data items generated by each node. The values in Table 5.2 for each class are the average latency violation amounts over all nodes in that class. Comparison of the latency violations of the nodes in Class2, which are the most demanding nodes with the highest sampling rates and strictest QoS requirements, shows how different methods are able to provide required latency services. Using FIFO-based or random item selection, the latency constraint for data items from nodes in Class2 are often violated. Moreover, a very poor DDR is provided for this class of nodes.



(a) Dynamic priority-based data selection



(b) Uniform random data selection

Figure 5.3: The average latency of data items over nodes c_3, c_4, and c_5 of each cluster (WBAN) with a time frame of more stringent QoS requirements using dynamic prioritization and random item selection mechanisms.

Simulation Setup2: Dynamic QoS Requirements

The second simulation setup aims to observe the behavior of the mechanism in a network with varying QoS requirements. The basis of the setup is the same as for the first simulation setup with the same node classes having the requirements shown in Table 5.1. However, in this case for some time periods, body nodes in Class3 decide to increase their sampling rate and switch to the same requirements as nodes in Class2 ($T_s = 3s,\ \mathcal{LC} = 20s,\ \mathcal{RC} = 0.8$). In a healthcare monitoring application, this may happen for certain patient conditions. Three WBANs in this experiment (C^1, C^2, and C^3) make this change in separate time frames of length 1000 rounds starting from simulation round 500, 1500, and 2500, respectively. The WBANs remain static in this experiment so that the reaction of the mechanism to the changes in the requirements can be observed separately from the effects of mobility.

Fig. 5.3 exhibits the average latencies of received items over three sensor nodes c_3, c_4, and c_5 of each WBAN using dynamic data prioritization and random item selection mechanisms. Fig. 5.3(a) reveals how the QoS requirement changes are tracked by our dynamic priority mechanism. For instance, nodes in Cluster1 switch to Class2 requirements between rounds 500 and 1500. The figure shows that latency values are considerably lower in this time duration for this cluster. The achieved DDR during this period is also higher than that in the normal situation. In contrast, random items selection strategy provides a very poor DDR in the time periods with more stringent QoS requirements and higher sampling rate (Fig. 5.3(b)). This is because random item selection mechanism does not take into account the sampling specification and QoS requirements of the data items.

Simulation Setup3: COPD Monitoring Application

The last set of simulations is done based on the deployment of a real healthcare application. The Roessingh Research and Development center performed an experiment for COPD patient monitoring. Patients suffering from COPD should be careful about the amount of activity they perform during a day and they have to distribute their energy over the whole day. The main goal of the experiment was to investigate the possibility of non-intrusive Activities of Daily Living (ADL) recognition by analyzing the information gathered by a sensor network. The patient's home is equipped with MyriaNed nodes. Fig. 5.4 shows the node placement and sensors type of the first floor of the house.

We use the deployment of this experiment for node placement on our simulation area as shown in Fig. 5.4 as a case study to test our proposed mechanism. There are 42 static nodes and one WBAN with five sensor nodes. Although there are several kinds of sensors in the static part, to simplify presentation of the results, we assume all static nodes to have the same requirements except for the accelerometers (s_{29-42} in Table 5.3) that have more stringent QoS requirements. We also have two classes of requirements for WBAN nodes. Table 5.3 presents the sampling period and QoS requirements of the four sensor classes that we used for our simulations. We assume stringent requirements for data from some of the WBAN sensors. ECG sensors (c_1 and c_2) of the WBAN have the highest possible sampling rate (equal to the TDMA frame length) with highest QoS

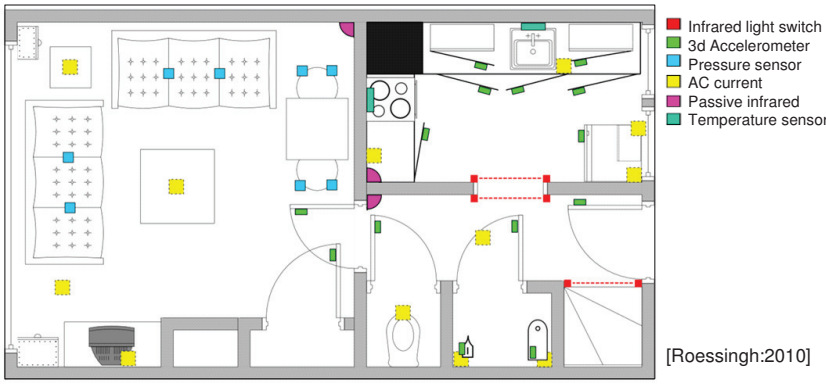


Figure 5.4: The floor plan and node placement in the COPD experiment.

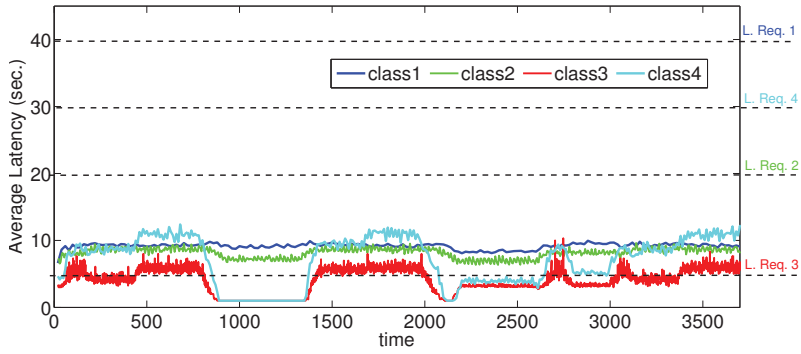
Table 5.3: QoS requirements of sensor nodes in COPD simulation setup

class	**nodes**	T_s	$\mathcal{LC}$ (*Latency req.*)	$\mathcal{RC}$ (*Reliability req.*)
Class1	s_i $(1 \leq i \leq 28)$	20 sec.	40 sec.	0.5
Class2	s_i $(29 \leq i \leq 42)$	5 sec.	20 sec.	0.7
Class3	c_1 , c_2	1 sec.	5 sec.	0.8
Class4	c_3 , c_4 , c_5	10 sec.	30 sec.	0.7

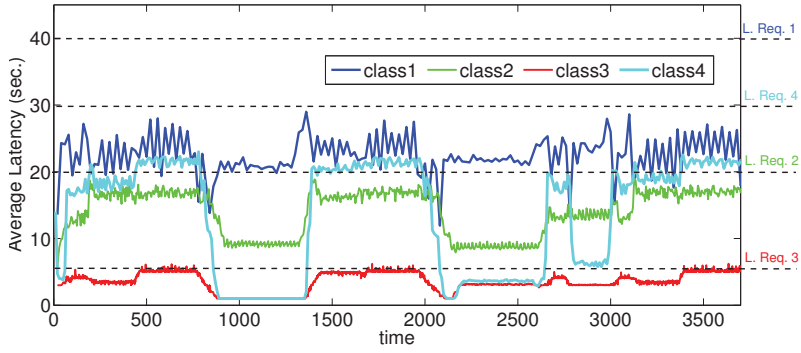
requirements. This is the most important information, which should reach the sink very fast. Other sensor nodes on the body (Class4) have less strict requirements.

Fig. 5.5 shows the average latency of received data items over all nodes in each class in a simulation of the setup for priority-based and random item selection strategies. Major variations in the latency values are caused by the variations in the relative hop-distance of WBAN nodes to the sink node due to the mobility of the patient. In the time frame between rounds 900 and 1400, the WBAN was one-hop away from the sink. So the latency is one second (one TDMA round) using either the random or the priority-based item selection.

The results obtained by performing priority-based item selection (Fig. 5.5(b)) show that the mechanism properly distributes the network capacity among data items according to the requirements of each class. In contrast, Fig. 5.5(a) reveals that the latency constraint of Class3 (which has the tightest requirements) is sometimes violated using the random item selection. It happens even though the achieved DDR value for this class is very low (38%). Again we emphasize that the latency values are just for the data items that reach the sink node. To gain a better understanding of the situation, consider the achieved DDR values shown in Fig. 5.6. Using random item selection, good latency and DDR values are obtained for Class1, which is not necessary. But the DDR value of high reliability demanding items of Class3, for instance, is quite low. By applying our priority-based item selection, the network resources have been distributed more in accordance with the requirements.



(a) Uniform random data selection



(b) Priority-based data selection

Figure 5.5: The average latency of received data items for each class of nodes in COPD simulation setup (Setup3).

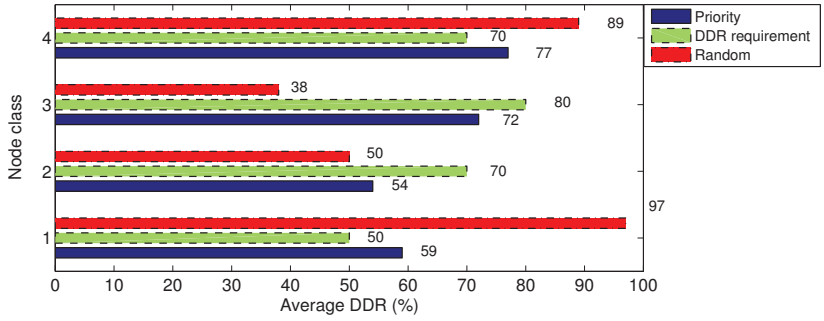


Figure 5.6: The requested and achieved DDR values in COPD simulation setup.

Table 5.4: QoS requirements and sampling periods of sensor nodes in the experiment setup ($1 \leq i \leq 60$, $1 \leq k \leq 4$)

class	**nodes**	T_s	$\mathcal{LC}$ (*Latency req.*)	$\mathcal{RC}$ (*Reliability req.*)
Class1	s_i (static nodes)	60 *sec.*	60 *sec.*	0.7 (70%)
Class2	c_1^k , c_2^k	3 *sec.*	15 *sec.*	0.8 (80%)
Class3	c_3^k , c_4^k	30 *sec.*	50 *sec.*	0.7 (70%)

5.5.3 Experiments

Besides the simulations, we perform real-world experiments using the indoor deployment presented in Chapter 4. A static sensor network of 60 nodes deployed in two floors of an office building and four WBANs each having 4 body nodes are used. Fig. 4.8 shows the deployment setup, but now static node s_{17} is the sink node. We consider a heterogeneous setup of sampling rates and QoS requirements inspired from specifications of health monitoring applications. Table 5.4 presents the specifications of the classes of the nodes in the network for the experiments. To have a comparison of the behavior of the data item selection mechanism, like in the simulations, we run experiments with data item selection using a uniformly random item selection, FIFO-based strategy, and our prioritization mechanism. The MCMAC protocol, presented in Chapter 4, and the gossip-based data dissemination, discussed in Chapter 2, are used for the experiments. The MAC and routing protocols are exactly the same for different experiments and the only difference is the item selection strategy. We tried to have similar mobility patterns for WBANs in different experiments. However, location and mobility of WBANs in different experiments cannot be guaranteed to be exactly the same.

The nodes in Class2 of the setup have the highest sampling rate and most stringent requirements. However, considering the network size and mobility of the WBANs, delivering at least 80% of the data samples with this rate of data generation seems to be challenging in this experimental setup. Table 5.5 presents the average DDR and latency of data items from the nodes in each class using different selection methods. The results show that none of the item selection mechanisms can satisfy the requirements of Class2. It means that the requirement may be too ambitious with respect to the capacity of the network setup. However, note that QoS management is not the goal of this work. As mentioned before, the goal here is to have a fair distribution of the service according to the individual QoS requirements of different data items. Table 5.5 shows that the achieved DDR using the prioritization mechanism are better distributed in accordance to the DDR requirements of each class. For instance, the random item selection mechanism provides a DDR of 92% for Class1 while the requirement is 70%. But the DDR value for Class2 is much lower (50%) while the requirement for this class is higher (80%). Using the data prioritization mechanism, the service is better distributed compared to that of the other item selection strategies.

As it is mentioned before, we can only calculate the latency of the arrived data items at the sink node. The data prioritization scheme provides slightly higher average latency for Class2 compared to the other item selection methods. However, other methods pro-

Table 5.5: Average DDR and latency over all nodes in each class in the experiments.

Class	$\mathcal{RC}$	*avg. DDR (%)*			$\mathcal{LC}$	*avg. Latency (sec.)*		
		Random	*FIFO*	*Priority*	*(sec.)*	*Random*	*FIFO*	*Priority*
Class1	70	92	46	84	60	18.8	96.6	57.5
Class2	80	50	38	61	15	4.2	4.5	4.9
Class3	70	75	53	71	50	9.1	13.9	28.7

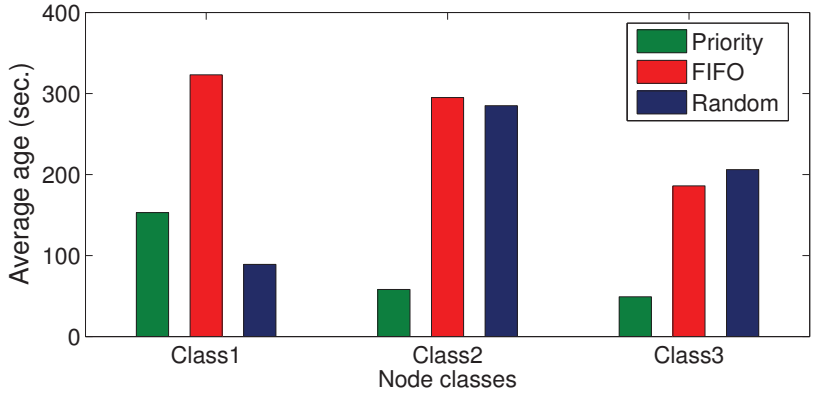


Figure 5.7: The achieved average age in the experiments for different node classes using different data item selection strategies.

vide very low DDR for this highly demanding class. When the data delivery ratio is low, the calculated latency might be high. Here we also consider the overall age of the data items at the sink node averaged over all nodes in each class. The definition of the age metric is given in Chapter 2. Fig. 5.7 depicts the achieved age using different methods in the performed experiments. Compared to the random selection, the priority-based mechanism provides slightly worse age for nodes in Class1, which are not demanding nodes. Instead, it considerably gains (lower age) for Class2, which includes more demanding nodes.

5.6 Discussion

5.6.1 Communication Overhead

To the aim of dynamically calculating priorities in all relaying nodes, we need to attach some information to each data item. This information includes the QoS requirements and the sampling period of the source node as well as the sample time of the item (for priority calculation in our application). The sample time of each data item was already built-in because it is essential data for the gossip-based data dissemination protocol. For

QoS requirements and sampling periods, both in simulation and experiments, we use a look-up table of the different classes. Then each source node only needs to attach the index of the corresponding class to its sampled data. All nodes have the same look-up table and are able to retrieve the values of QoS requirements and sampling periods with the index attached to a received data item.

In all our simulations and experiments, we only need to add two bits to specify the class. Considering that the length of each data item in the MyriaNed packet structure is eight bytes, we have a communication overhead around 3%. Such a low overhead is quite acceptable to gain the required QoS differentiation. However, the overhead might be different according to the number of possible classes and also the other information that we need to add to the data items in a certain application.

5.6.2 Robustness

The core idea of using a gossip-based data dissemination is to provide robustness through exploiting redundancy in the network. The gossiping mechanism discussed in Chapter 2, uses random item selection to provide a statistically equal chance for each data item to be forwarded in all directions in the network until it reaches the desired destination (sink). In this work, we consider scenarios with heterogeneity in the QoS requirements, besides the built-in heterogeneity and dynamics in WSNs. We propose to use a dynamic data prioritization instead of purely random data selection. Our simulations and experiments show that our mechanism performs better in distributing the network bandwidth according to QoS requirements of different data items. However, the impact on robustness of the network against unwanted events such as node failures and network disconnections is not studied in this work. As these events are sporadic and unpredictable, evaluating the robustness of different mechanisms is a challenging issue. This is considered as a complementary evaluation for the proposed mechanism that is considered as a follow-up for this work.

5.6.3 Retransmission Control

When a data item gets a higher priority compared to the other items, it is included in the transmission packet. However, there is a chance that the transmitted packet is lost due to causes such as interference or collision. A specific strategy then may be exploited for retransmission of already transmitted data items. For instance, a mechanism for retransmission control is presented in [15] that takes the quality of outgoing wireless links into account to find a statistically optimum number of times that a packet needs to be retransmitted to provide a certain probability of reception by the receiver node.

If the routing protocol uses a specific retransmission mechanism, then it may be integrated with the data prioritization mechanism. For the simulations and the experiments presented in this chapter, we gave the lowest priority (zero) to the data items that have been transmitted. They may still be retransmitted if the other items also have such low priority. Integrating our data prioritization mechanism with a smart retransmission mechanism based on the quality of the outgoing links is considered as future research on this topic.

5.7 Summary

Health applications show to have various kinds of spatial heterogeneity and temporal variations of QoS requirements of data samples from different sensor nodes. In this chapter, a dynamic data prioritization mechanism is proposed for data dissemination in wireless sensor networks (WSNs) with heterogeneity in the Quality-of-Service (QoS) requirements and sampling specifications among different wireless nodes. The goal is to distribute network bandwidth among the data items according to their relative QoS demands. The mechanism is specifically vital for multi-hop WSNs with high data loads. The method also supports time-varying QoS requirements, which is useful for multi-scenario applications such as health applications with emergency and normal scenarios.

In the proposed mechanism, the priority values are calculated dynamically in every node on the routing path according to the QoS requirement of each data item and the history of the data item. This way, mobility of nodes is implicitly taken into account. The mechanism can be used for data dissemination and routing using various kinds of data propagation protocols. We integrated it with the gossip-based protocol for a health application scenario as a representative case study. Extensive simulations with several setups and real-world experiments have been performed to observe the behavior of the data prioritization mechanism. The results clearly show that using dynamic data prioritization, the data items with more stringent QoS demands receive better service compared to the service provided for less demanding items.

Chapter 6

Wireless Body Area Networks

6.1 Overview

Recall from Chapter 1 that we discussed two different paradigms for WBAN communication, namely flat and gateway-based architectures. In Chapter 4, we focused on communication in networks with a flat architecture in which all body sensor nodes directly and independently communicate with the surrounding static (ambient) network. The main goal was to efficiently deliver body data to the static network considering the cluster mobility of WBANs. In Chapter 5, the focus was on QoS differentiation in dissemination of body data as well as ambient data through the static network to be delivered to the sink node(s). For this mechanism, the type of the architecture of WBAN communication makes no difference. This chapter focuses on communication of the body sensor nodes within the WBAN, when a gateway-based architecture is used in the network for WBAN communication. Body data generated by all body sensor nodes is collected by a gateway node on the body. Then the gateway communicates with the surrounding network. The goal of such intra-WBAN communication is to efficiently deliver data from all nodes in the WBAN to the gateway taking into account the special characteristics of WBANs. Stringent power constraints, short transmission range of the body nodes, time-variant quality of wireless links, and high mobility are some characterizing features of these networks.

In this chapter, we first review the existing approaches in designing protocols for intra-WBAN communication. These can be divided into two classes, namely those following a *star* network architecture and those that are variants of *multi-hop* architectures. Then, to address the challenges in WBAN communication and considering typical requirements in the target applications, we develop a hybrid mechanism for intra-WBAN communication with the main objectives of robustness, power efficiency, and simplicity. An On-Demand Listening and Forwarding (ODLF) mechanism is proposed that automatically changes from a star strategy to a multi-hop architecture and vice versa according to the network conditions. When some node cannot reach the gateway node, the mechanism uses a

gossip-based data forwarding scheme to deliver its data to the gateway through a multi-hop communication.

The proposed ODLF mechanism is evaluated by deploying several wireless motes on a body and performing various experiments. The results confirm the robustness and improved behavior of this protocol in comparison with existing fixed protocol architectures. Experiments show that this protocol gives a good trade-off in terms of performance metrics and energy consumption and can be a generic solution for communication in many WBAN application scenarios. This chapter is based on previous publications [57, 62]. Moreover, a demonstration of our experimental setup and the online QoS evaluation system has been presented in [61].

This chapter is organized as follows. The next section discusses the special characteristics of WBANs that make them different from ordinary WSNs. Sec. 6.3 gives a review of two general approaches in protocol design for WBANs. Our proposed on-demand data forwarding (ODLF) mechanism is presented in Sec. 6.4. The experiments, comparing several protocol design approaches, are explained in Sec. 6.5.

6.2 WBAN Characteristics

Several experimental studies including our own experiments using wireless nodes show that WBANs have specific characteristics that differentiate them from typical WSNs. Thus special attention is required for designing appropriate communication protocols for such networks. In this section we point out some special characteristics of these networks.

6.2.1 Energy Consumption Constraints

Body sensor devices are expected to develop into very tiny, light-weight, and even flexible nodes to be wearable during normal daily life. These physical shape requirements limit the energy source of the wireless body nodes. The battery of on-body sensor nodes needs to be very small to follow the size requirements of the whole device. Moreover, some biological nodes such as some implantable sensors do not even have a battery. In such applications, sensor devices use energy-scavenging from body sources [82] (e.g., producing energy from body temperature, body motion, or heart beat [110]). Consequently, the total amount of consumable energy for both wireless communication and computation is strictly limited. Therefore, the energy consumption constraints of body sensors are tighter than in many other wireless sensor applications. In all WBAN design steps, from designing the physical layer of sensor devices to communication protocols and assigning computation tasks, energy consumption should be carefully limited and optimized.

6.2.2 Quality of Wireless Links

Experimental observations show that the quality of wireless links between on-body sensor nodes is low and time-variant. Experiments using CrossBow TelosB [102] sensor nodes presented in [64] and [63] show that PRR between some on-body nodes is lower than

60% for 46% of the experiments. The status of wireless links in experiments of [72] varies in different postures. Moreover, even within a given posture, link quality may vary and have intermittent connections or failures. Experiments presented in [18] show the periodic variation between low and high quality of the link between two nodes placed on the wrist and belly during walking of the person wearing the nodes. We also performed several experiments by deploying several MyriaNed nodes on a body in different postures to get insight into the behavior of the wireless links. The experimental setup and results are presented in Sec. 6.5. These experiments confirm the link unreliability and time-variance issue in WBANs. In the following, we mention prominent some causes for such low quality of links.

Low-power RF Radio: The main energy consuming block of wireless sensor devices is their RF radio transceiver. To follow very tight energy consumption constraints of body sensor nodes and to enlarge their lifetime, ultra-low power RF devices are used in WBANs. Moreover, to provide very small size features and convenience requirements, a big RF antenna is not affordable. Very small on-board or even on-chip antennas are used for body sensors. A low power radio transceiver with a tiny antenna leads to a very short transmission range for these nodes. The literature describes several low-power body sensors with a transmission range below one meter ([27, 85, 92]).

Lossy communication channel: Water severely absorbs some frequency bands in the spectrum of the RF waves. The 2.4 GHz ISM band is one of those frequency bands. However, this frequency band is one of the carrier frequencies suggested for WBANs by IEEE 802.15.6 [11]. 50% to 65% of the human body consists of water. Thus, the human body can severely shadow the RF signal and presents a high attenuation wave propagation channel. Several empirical investigations show that the propagation loss around and in a human body is considerably high ([31, 39, 77, 81, 88, 106]). The path-loss coefficient value lies between 4 and 7, whereas its value is 2 in vacuum space and around 3 in air [74]. This makes the links between body nodes unreliable and further limits the transmission range. Moreover, radio channel in WBANs is shown to be subject specific [108] and its characteristics vary over time due to mobility and posture changes.

External interference: The 2.4 GHz ISM band is an unlicensed frequency band that may simultaneously be used by other commercial devices in the environment of a WBAN. IEEE 802.11 (WLAN) [9] and IEEE 802.15.1 (Bluetooth) [8] are examples of coexistent users of this frequency band. Microwave ovens also use this frequency band. This leads to likely external interference on WBAN links. In [32], an experimental investigation is done to investigate the impact of WLAN interference on WBANs. It concludes that when body nodes use a very low transmission power, the external interference can cause substantial packet losses in WBANs (more than 50% transmission failure during the experiments when transmission power of -25dBm is used by the body nodes).

6.2.3 Mobility

The human body is mobile and can be in several postures. The movements of the sensor nodes deployed on different positions of the body changes the distance between the nodes. A more important effect of such movements is that the channel coefficient substantially changes, depending on the fraction of the distance between two nodes that the RF waves should pass through or along the body. The wireless link between two nodes that are far away from each other, but can see each other, may be of good quality, whereas two close nodes cannot reach each other because of the positioning of the body between them. Therefore, an important factor influencing the quality of the links in WBANs is the fraction of the distance between two body nodes that RF waves propagate through or along the human body.

The mobility of nodes causes frequent changes in the quality of links in WBANs. Moreover, postural changes of the human body may frequently change the network topology. Thus no assumptions can be made for the node neighborhood based on the relative node positions. The network protocol should be robust against such frequent link quality fluctuations and topology changes.

6.2.4 Network Size

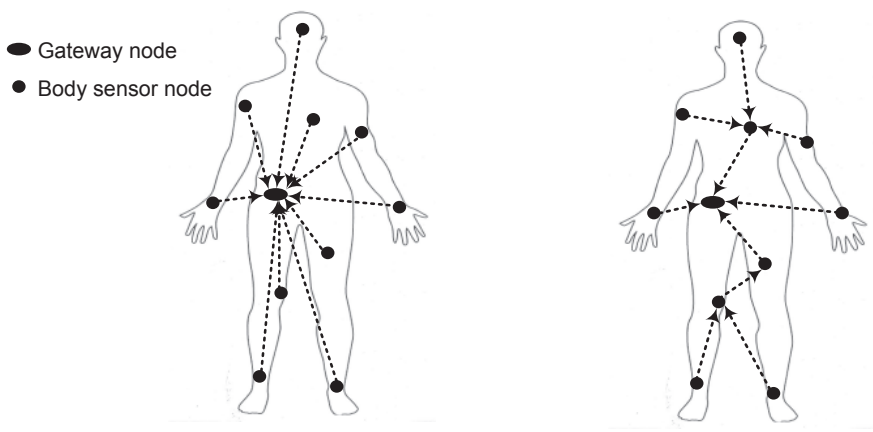
Among the above challenging properties of WBANs, one relaxing property of WBANs is that the number of sensor nodes on a body is limited. Deploying many nodes on the body may cause inconvenience for daily life. So the problem of scalability of protocols is not a challenging issue in WBANs. This allows us to make some simplifying assumptions. For instance, we may assume that the gateway node has information about sampling and transmission rate of the body nodes within the WBAN. Dedicating a unique transmit slot to any body node in the TDMA scheduling without the need for RF channel reuse is another decision that can be made considering the low number of nodes in typical WBANs.

6.3 WBAN Protocol Design Approaches

Literature on WBANs describes several MAC and data propagation protocols for communication within WBANs. Although differing in the details of the protocols, they all assume either a star topology or a multi-hop architecture for delivering the sensed data to the gateway node. These two design approaches are discussed in this section.

6.3.1 Star Architecture

The first choice for designing a WBAN has been a star (one-hop) architecture (Fig. 6.1(a)). In this architecture, the sensor nodes are supposed to send information directly to the gateway node placed on the body. In [54], [65], and [70], for instance, MAC protocols are proposed for WBANs assuming an underlying star architecture. The IEEE 802.15.6 [11] WBAN standard assumes a star network topology with one gateway in each WBAN.



(a) Star (one-hop) architecture (b) Multi-hop communication

Figure 6.1: Common approaches for intra-WBAN communication.

Considering the short distances between nodes in a WBAN, a star architecture seems to be a reasonable option that in general is simple to implement with an overall low latency because of direct links to the gateway. Moreover, nodes in a star architecture are independent. For instance, in the case of a node failure, other nodes continue to work as normal. However, this architecture suffers from low DDR and insufficient reliability caused by low and time-variant quality of wireless links. Further, the power constraint and short transmission range of typical on-body sensor nodes, and the severe wave propagation loss through the body limit the effectiveness of such architectures for many applications. Our experiments discussed in Sec. 6.5 show that in certain postures of the body, some nodes are disconnected from the gateway node, even if they use their highest transmission power. This shows that a star architecture cannot be sufficiently reliable in all applications.

6.3.2 Multi-hop Communication Protocols

The alternative architecture for WBANs is multi-hop communication (Fig. 6.1(b)). In such networks, nodes are not required to send their data directly to the gateway node. Instead, they send their data to certain neighbors in range according to the underlying routing structure hoping that their data reaches the gateway in one or more hops. A tree-based multi-hop protocol is proposed in [49] in which the routing spanning tree is set up autonomously to route data from the sensor nodes to the gateway. A common problem of every tree structure in WSNs is that in the case of a node failure or node movement, the tree has to be reconstructed. Although a method for tree reconstruction is presented in [49], it is mentioned that only low mobility is supported. A posture change in a human body will likely change the body network topology. Moreover, frequent posture changes demand a frequent tree reconstruction, which is energy- and time-costly.

In [71], relative location-based forwarding is exploited in which the nodes always forward their packets to other body nodes that are known to be closer to the gateway. Every node is programmed with the distance of all nodes in the WBAN to the gateway. Having this information, each node knows which nodes are closer to the gateway. However, considering the postural changes of the human body, the relative distances of the body nodes to the gateway node are prone to change. Moreover, as it was mentioned, a major factor influencing the link quality is the body shadowing and the distance may have less influence on the link behavior.

A probabilistic routing mechanism is proposed in [72]. Each node continuously measures the quality of its link to other nodes by calculating a stochastic metric called link likelihood factor (LLF). All nodes broadcast their calculated LLF. Therefore, all nodes are informed about the other links' quality and are able to calculate the overall quality (cost) of their possible paths to the gateway node. Each node then tries to send its data to a neighbor with the best link (highest LLF) to the gateway. The first drawback of this protocol is the reliance on symmetric links. Node c_i uses the reception rate of the hello messages transmitted periodically by another body node c_j to estimate the quality of its outgoing link towards node c_j. Assuming symmetric quality for low power wireless links is typically unrealistic [56, 87]. The second drawback of this approach is the data exchange overhead imposed for detecting the best connection path. Because of frequent topology changes and high link quality variations, the estimated best forwarding paths may not remain valid for a long time and frequent adaptation is necessary. It means that calculating LLF values and broadcasting them should be done often enough to follow dynamics in the network. These frequent costly tasks need to be done by the energy-constrained body nodes.

Multi-hop protocols are shown to have in general a high end-to-end DDR [64]. The other advantage is that the body nodes are relaxed from the requirement of reaching a certain node (gateway). Thus they may be equipped with very low-power radios. Considering the low transmission range of typical body sensor nodes and the low wireless channel quality, a multi-hop architecture can be beneficial, and in some cases it is the only possible option. On the other hand, multi-hop protocols are more complicated and the network may suffer from longer latencies. Experiments in [64] and [63] explore the trade-off between using a star and a multi-hop architecture, highlighting their respective performance characteristics. There is no solution that is optimal for all applications of WBANs because of different constraints and requirements.

To deal with the issues of link unreliability and overhead of routing structure reconstruction in existing multi-hop protocols, we first investigate exploiting a gossiping strategy for data dissemination within the WBAN. We exploited an adapted version of the gossiping mechanism [29], which was explained in Chapter 2. It does not use any specific routing structure and does not make any assumption about the relative position of nodes or link properties (such as DDR or symmetry). This makes the protocol robust against such WBAN challenges as high node mobility (posture changes) and low quality wireless links. Every node listens to all other nodes in the WBAN to possibly receive packets from them. If a packet is received, data items are stored in a memory cache and are subsequently forwarded hoping that the gateway receives them. We also fortified this

Table 6.1: A qualitative comparison of different network architecture approaches for WBANs.

Criteria	**Star architecture**	**Multi-hop architecture**			
		Tree-based routing	**Full gossip**	**Probabilistic routing**	**ODLF**
Implementation complexity	simplest	complex, tree structure construction and maintenance	simple	rather complex	rather simple
Energy consumption (radio activity)	low, if no acknowledgement is performed	high, unnecessary data forwarding when nodes can reach the gateway	high, always listen to all nodes and forward packets	high, packet exchanges for maintaining best routes to the gateway	depends on quality of links, low when nodes can reach the gateway properly
Latency of delivered packets	low, always one hop	high, always multi-hop	varying, low in networks with high-quality links	varying, depending on the link quality	varying, low (one hop) when it is possible
Robustness against node failures	robust, other nodes work as normal	not tolerated, tree reconstruction is required	robust, no dependencies between nodes	best paths should be updated	robust, no dependencies between nodes
Mobility support	always one hop, no route maintenance needed	frequent tree reconstruction is needed, which is time- and energy-costly.	native, no specific routes, no assumption about node positions	by frequent updates of link quality estimates and best paths	native, no specific routes, no assumption about node positions
Overall end-to-end DDR	low, some nodes may not directly reach the gateway	rather high, short distances between neighboring nodes in the tree	high, all nodes always receive and forward other node data	high, forwarding by selecting the paths with best quality links	high, forwarding only for nodes in trouble
Robustness against long-term outages	no support, which may cause long burst packet losses	tree should be adapted, some burst packet loss may happen in this time frame	native, no effort to detect outages	best path to the gateway should be continuously updated by the nodes	native, outages are detected and data forwarding and listening is adapted

mechanism with a dynamic transmit power adaptation mechanism in order to maintain required connections and optimize transmit energy consumption. Although using such a gossip-based routing mechanism provides a good robustness for the network, it is not optimized from the power consumption point of view, as the nodes always listen to other nodes and always forward information both of which might not always be necessary. This mechanism was published in [57]. For the rest of this chapter, we refer to this intra-WBAN communication mechanism as Full Gossip (FG). Our ODLF mechanism as presented in this chapter builds upon this approach.

Table 6.1 summarizes this discussion by presenting a qualitative comparison of different approaches for intra-WBAN communications in terms of implementation complexity, and various performance metrics. It also includes our FG and ODLF mechanisms.

6.4 ODLF: On-Demand Listening and Forwarding

We propose using a hybrid approach to develop a *simple*, *robust*, and *power-efficient* protocol for intra-WBAN communication taking benefit from both star and multi-hop approaches. In this section, we first present the overall description of this mechanism. We then continue with a discussion on the characteristics of disconnections from the gateway in WBANs. Then we explain all parts of the mechanism step by step.

6.4.1 ODLF Principle Review

In the ODLF mechanism, the body sensor nodes try to send their data directly to the gateway node, like in a star topology. The gateway node continuously estimates the quality of all its incoming links. Using this estimation, the gateway detects the body nodes that do not have a sufficiently good connection. The gateway then distributes a short summary to inform body nodes about the nodes that cannot properly reach the gateway.

The body nodes receive the information about the disconnected nodes from the gateway and listen to them. If they receive any packet from disconnected nodes, they use a store-and-forward scheme to propagate the received data items, aiming to deliver them to the gateway. The network will have a star topology when all nodes have a good connection to the gateway. However, if some nodes do not have a proper link to the gateway, the topology will automatically change to a required level of multi-hop communication. A gossiping strategy like the one mentioned in Chapter 2 for MyriaNed nodes is used for data forwarding. Therefore, even for on-demand multi-hop data dissemination, no specific routing structure is required to be established and maintained. This keeps the network protocol simple and robust.

6.4.2 Outage Characteristics in WBANs

We experience two kinds of disconnection in WBANs. The first kind of outages are rather short term and caused by fading due to movement of the nodes or a temporary interference. This frequently happens in mobile postures such as walking. An empirical measurement for outdoor setting in [70] reveals the periodic fluctuation of RSSI of several links while walking. It is shown that the RSSI fluctuation frequency matches the walking step frequency (i.e., 1.2 steps per second in [70]). Moreover, the IEEE 802.15.6 working group [18] has reported the results of extensive indoor experiments, proposing a two-state channel model for walking in a WBAN. The transmission of a node may happen while the RSSI has its own high peak and so it succeeds. On the other hand, the packet may be transmitted while the link is in its weak state and so fails. Therefore, we may see some transient failure or success in packet transmission of a node. If a node has a good link to the gateway, this kind of short outages can be combated using retransmission of the packets by the source node itself.

The second kind of outages, which are specially the target of our on-demand forwarding mechanism, are longer term outages. Shadowing caused by posture changes and also movements can bring a link to a situation in which the node cannot reach the gateway for a long period of time. We observed many cases during our experiments in which some nodes cannot send any packet to the gateway for the whole time during that the same type of posture was kept. This kind of outage is especially very problematic since there may be no data reception from a node for minutes or even longer. This, in turn, may lead to a serious failure of the application.

6.4.3 Gateway Measurement and Feedback

As the core of the on-demand data forwarding, the gateway measures a metric based on which it reports the nodes with a poor link to the gateway. There are two points in such measurements. First, the metric should differentiate short and long term outages so that the forwarding mechanism is initiated only when a node is in a long term outage state. Second, it is important to detect this situation quickly to minimize the amount of data loss of the node with a poor link.

Detecting the nodes disconnected from the gateway to trigger data forwarding support consists of three steps. First, the gateway uses a specific metric to quantify the quality of its direct incoming links. In the second step, the gateway uses the computed link quality values to detect the disconnected nodes according to individual requirements and specification of each node in the WBAN. Finally, the gateway informs all other nodes about the nodes disconnected from the gateway.

Link Quality Estimation (LQE)

In the first step, we need to estimate the quality of the wireless links in the WBAN. We first review a classification of different LQE techniques and then present the method that we use in the ODLF mechanism. Techniques for estimating the quality of wireless links in WSNs are categorized as *hardware-based* and *software-based* methods [14]. Link Quality

Indicator (LQI), Received Signal Strength Indicator (RSSI), and Signal-to-Noise Ratio (SNR) are hardware-based LQE methods provided by many radio devices. The hardware-based link quality values are directly given by the radio device and no further computation is necessary in software. However, these techniques suffer from some limitations. First, the provided values are only for successfully received packets and the lost packets are not taken into consideration. This may lead to overestimating the link quality. Second, the hardware-based LQE techniques are only suitable for classifying the links as very good or very bad and are not accurate enough to provide a fine-grained estimation of the link quality [28]. Another fact that further limits the use of hardware-based techniques is that not all radio chips are equipped with a link quality estimator (e.g., Nordic nRF24L01 [4]).

Software-based LQEs are done in software according to the ability of the links to deliver the whole packets (data packets or beacons), and do not rely on hardware support. In general, software-based LQE methods are able to characterize the link quality with a better precession [14]. A software-based link quality estimation can use the existing data packet transmissions, without extra beacon transmissions, if the packet transmission rate is high enough. For very low transmission rates, beacon packet transmissions may be used for link quality estimation.

In the ODLF mechanism, we use a software-based LQE technique, which is based on the packet reception ratio of the wireless links from body nodes to the gateway. PRR is a commonly used metric for estimating the quality of the wireless links that considers a certain history of the link status. It provides a stable estimation [14], but suffers from slow reaction to the link quality dynamics. We use the LLF metric proposed in [72], to increase the reactivity of PRR. In the following, we present and discuss this quality estimator.

Assume that $C = \{c_i : 1 \leq i \leq M\}$ is the set of M sensor nodes deployed on a body. One of these nodes, say c_M, is the gateway. Recall from Chapter 2 that every node c_i samples a specific signal with period of $T_s(c_i)$ (sampling period) and transmits packets every $T_{tx}(c_i)$ units of time (transmission period). The gateway node is assumed to know the sampling period of every node in the WBAN. Thus, the gateway knows when each node transmits a packet and is able to measure the link quality according to the packets that it receives from each node. Let $L^{t,k}_{i\to M}$ be a logical value (0 or 1) that shows whether the k^{th} packet transmitted by node c_i before/at time t has been directly received by the gateway node c_M. Note that here we only consider the status of the direct link between node c_i and the gateway c_M. The value of PRR for c_i at time t, denoted by $P^t_{i\to M}$, over the last h transmission rounds is then calculated as follows.

$$P^t_{i\to M} = \frac{1}{h}\sum_{k=1}^{h} L^{t,k}_{i\to M} \qquad 1 \leq i < M \tag{6.1}$$

The choice for parameter h depends on the characteristics of the movement and postural changes. Considering 10 seconds for the duration of posture changes seems to be reasonable[1] and so a value of $h = 10/T_{tx}(c_i)$ is set. As the value of $P^t_{i\to M}$ gives the

[1] A value between 7 and 14 is suggested in [72] to be used for the length of the averaging window.

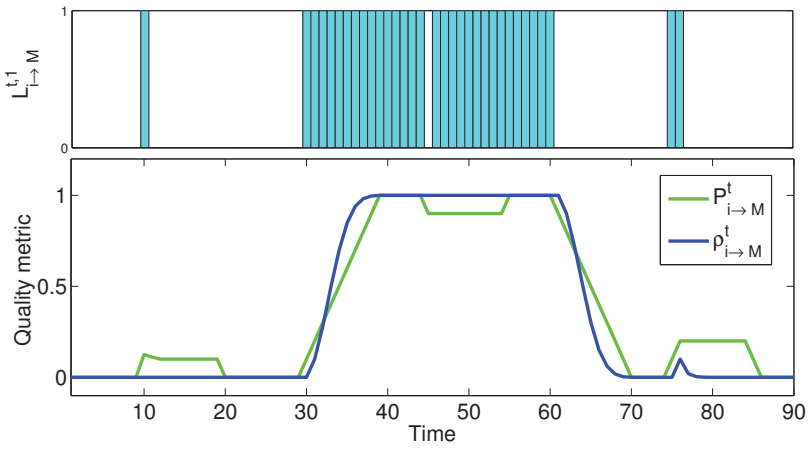


Figure 6.2: Packet reception ratio ($P^t_{i\to M}$) and the LLF ($\rho^t_{i\to M}$) metric for a sample link.

average status of the link from node c_i to the gateway in a time window, it adheres the same importance to all previous samples. To adapt accurately to new circumstances, the link history should be taken into account and the metric should react quickly to dynamics influencing network topology such as mobility and posture changes. If the metric value is increased late, information from the corresponding node may be lost for a while because other nodes have not been informed about listening to that node. Decreasing it late does not lead to information loss, but causes unnecessary listening for nodes that were participating in forwarding the packets of that node. On the other hand, the metric should be stable against short term outages.

The LLF metric, denoted by $\rho^t_{i\to M}$, tries to increase the reactivity of the PRR metric to non-transient changes of the quality of links considering the link status record. Eqn. 6.2 gives the calculation of this metric.

$$\rho^t_{i\to M} = \begin{cases} \rho^{t-1}_{i\to M} + (1 - \rho^{t-1}_{i\to M}) \times P^t_{i\to M} & L^{t,1}_{i\to M} = 1 \\ \rho^{t-1}_{i\to M} \times P^t_{i\to M} & L^{t,1}_{i\to M} = 0 \end{cases} \tag{6.2}$$

If the gateway successfully received the last transmitted packet by node c_i (the first case in Eqn. 6.2), the metric is increased towards 1 with a rate proportional to the historical PRR of this link ($P^t_{i\to M}$) and the difference of its previous value $\rho^{t-1}_{i\to M}$ to the maximum value (1). If the link from c_i to the gateway has shown a good connection in the history, $P^t_{i\to M}$ is relatively high. Consequently, $\rho^t_{i\to M}$ will converge to 1 exponentially with a very fast rate. On the other hand, if the gateway does not have a good history of receiving from c_i, the value of $P^t_{i\to M}$ is low. Thus, the last successful reception is

However, the exploration of the design space and the trade-offs made by this parameter to find its optimum value for the ODLF mechanism is considered as a future work.

considered as a likely temporary connection and so $\rho^t_{i\to M}$ will increase with a lower rate. In case c_i has not succeeded to reach the gateway in the last transmission ($L^{t,1}_{i\to M} = 0$), the metric is decreased, again with a rate proportional to the historic PRR of the link. If $P^t_{i\to M}$ is high, the last link failure is supposed to be an incidental disconnection and the metric $\rho^t_{i\to M}$ decreases with a low rate. The main idea behind this is that if the link has shown a good record, incidental disconnections do not decrease the metric too much and vice versa.

Fig. 6.2 illustrates the reaction of the PRR and the LLF metric $\rho^t_{i\to M}$ for a sample link.Until time 30, the link quality is very poor. Only one instantaneous reception happens at time 10. Here the PRR slightly increases, but the value of the LLF metric dose not change. This illustrates how the LLF metric tolerate very transient connections. At time 30, the link becomes connected with a good quality because of a posture change, for instance. Here, The LLF factor reaches its maximum value faster that PRR. Again, a transient packet lost (at time 45) does not change the value of LLF. At time 65, the link turns to a bad connectivity. The LLF metric reacts faster the PRR to such change. This example shows how the LLF metric outperforms PRR in ignoring transient link failures or successes and reacting faster to long-term connection changes.

Disconnection Detection

Based on the link quality metric $\rho_{i\to M}$, the gateway decides whether node c_i needs data forwarding help, by comparing its link quality with a given threshold ℓ_i. The value of the threshold is given individually for each node in the WBAN according to the nature of the signal being sampled by node c_i, and its required level of QoS such as reliability. Next we discuss how to decide on these thresholds.

The value of the threshold influences the performance of the mechanism and should be set carefully. Higher thresholds impose more power consumption overhead to the nodes as they have to listen to more nodes. It also affects the efficiency of the data forwarding mechanism. When nodes have to listen to more nodes, they will have more items in their data cache waiting for transmission. So the probability of being selected from the data cache decreases. It means that we may be sacrificing some really demanding items for propagating items that might already have reached the gateway. On the other hand, a very low value for the thresholds boils down to the direct data delivery to the gateway. Thus, setting the right thresholds is crucial for an optimal working condition for the WBAN.

Disconnection of a node from the gateway depend on the sampling rate and DDR requirement for that node. *A body sensor node c_i is considered disconnected from the gateway if it statistically cannot satisfy the required data delivery ratio $\mathcal{RC}(c_i)$ by sending its packets directly to the gateway node within the available transmission budget.*

Body nodes may use a retransmission scheme to have better DDR against short term outages. The node c_i generates samples every $T_s(c_i)$ seconds and transmits a packet with period of $T_{tx}(c_i)$. So a data item generated by node c_i has on average $\frac{T_s(c_i)}{T_{tx}(c_i)}$ rounds of opportunity to be transmitted directly by c_i itself (assuming that $T_s(c_i) \geq T_{tx}(c_i)$). Let X_i be the (stochastic) number of required transmissions of a single data item of c_i

to be received by the gateway for the first time. We assume that X_i has a geometric distribution. This assumption is confirmed to be reasonably accurate by the result of experiments that we have performed using wireless nodes. To statistically satisfy the packet delivery requirements of node c_i denoted by $0 < \mathcal{RC}(c_i) \leq 1$, the probability of X_i being less than the number of available transmission rounds should be less than the reliability requirement $\mathcal{RC}(c_i)$.

$$\mathbf{P}(X_i < \frac{T_s(c_i)}{T_{tx}(c_i)}) > \mathcal{RC}(c_i) \tag{6.3}$$

The probability of a successful transmission from c_i directly to the gateway is approximated by the PRR of the link ($P_{i \to M}$). Considering the cumulative distribution function of the geometric distribution, we come to Eqn. 6.4.

$$1 - (1 - P^t_{i \to M})^{\frac{T_s(c_i)}{T_{tx}(c_i)}} > \mathcal{RC}(c_i) \tag{6.4}$$

Solving that, we obtain the threshold ℓ_i for the quality of the link from node c_i to the gateway, as follows.

$$P^t_{i \to M} > \underbrace{1 - (1 - \mathcal{RC}(c_i))^{\frac{T_{tx}(c_i)}{T_s(c_i)}}}_{\ell_i} \tag{6.5}$$

At run time, a link with a PRR less than this threshold does not statistically satisfy the reliability requirement. This is the point that the on-demand listening and forwarding mechanism is triggered to help the node with insufficient link quality to the gateway (disconnected node). In our protocol, we compare the value of the LLF metric (Eqn. 6.2) with this threshold.

Announcing the Requested Set

Let $\Psi^t \subset C$ be the set of nodes without a sufficiently good link to the gateway at round t, called the *requested set*.

$$\Psi^t = \{c_i \in C \mid \rho^t_{i \to M} < \ell_i\} \tag{6.6}$$

To activate on-demand forwarding, the gateway includes a bitmap consisting of $M-1$ bits in its packets in which the i^{th} bit reflects whether $c_{i+1} \in \Psi^t$ (the bitmap is indexed starting from 0). Note that in the implementation of the mechanism, some hysteresis is applied to prevent too frequent switches when the value of the quality factor is around the threshold. We add c_i to the requested set when $\rho_{i \to M}$ drops below threshold ℓ_i. In contrast, for excluding c_i from Ψ^t, the quality factor should go higher than $\ell_i + \Delta\ell$. We used value $\Delta\ell = 0.2$ in the implementation, after trying some initial experiments with different values of the threshold. Using such a hysteresis, the node is kept in the requested set while its link quality to the gateway is around the threshold.

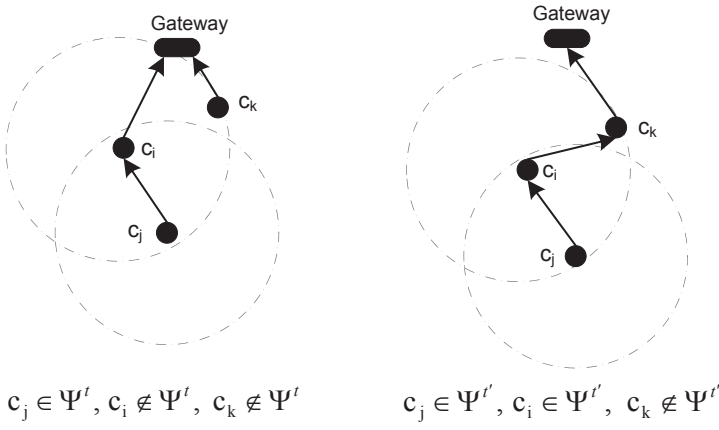


Figure 6.3: An illustrative example of the on-demand forwarding mechanism.

6.4.4 Listening Schedule

Each node transmits its data in the assigned transmit slots hoping that the gateway receives the packets. On the other hand, nodes receive a bitmap, stating the requested set Ψ^t at each round, which represents the disconnected nodes from the gateway[2]. Then, each node starts listening to the slots dedicated to the nodes in Ψ^t. Subsequently, if they receive any data from those nodes, they participate in forwarding that data.

Note that a node c_i tries to participate in data forwarding for the nodes in the requested set regardless of whether node c_i itself has a direct link to the gateway. This provides a multi-hop structure (not only 2-hops) for data propagation within the WBAN. For example, suppose that node c_j cannot reach the gateway due to large distance or low transmission range while node c_i is in its radio range. So if c_i listens to c_j, it receives and forwards the data of node c_j. If node c_i itself has a direct link to the gateway, then the packets of c_j reach the gateway in two hops (Fig. 6.3, left side). Otherwise, the same procedure happens for all propagated information from node c_i by another node, say c_k (Fig. 6.3, right side). Note that considering the typically rather small network size and deployment area, the multi-hop paths will not be long (at most three hops in any of our experiments).

With such listening mechanism, the number of listening slots in each frame is determined by the number of nodes in the requested set. When all nodes have good connection

[2]Note that all body nodes listen to the gateway to receive this information. The gateway node usually uses a more powerful radio. Thus we assume that the links from the gateway to the body nodes are of good quality. However, intermittent failures of this links does not lead to a failure of the mechanism. Later in this chapter we explain what a body node does if it has not received the bitmap from the gateway in a round.

to the gateway, the requested set is empty and nodes do not listen to any other nodes in the WBAN (except the gateway). If some nodes do not have good enough links to the gateway, other nodes listen to them to forward their data.

Listening to the entire requested set can be thought of as a conservative strategy to provide as little as possible data loss in the case that some nodes have no proper link to the gateway. However, more aggressive schemes can also be exploited here to further reduce power consumption of the nodes. The point is that when the quality of link from node c_j to c_i is very poor, it may make no sense that node c_i consumes energy for listening to c_j with only a low chance of success. So c_i can maintain a *listening subset* of the requested set at any point of time from which it can hear well. Such a subset should be carefully updated to include nodes that are still in the requested set and whose link to c_i becomes good, because of movement for instance.

To better understand this potential case, suppose that c_j is in the requested set ($c_j \in \Psi^t$), but c_i cannot hear c_j in the current posture and decides to exclude it from its listening subset. At a later time, a change in the posture brings c_i into the transmission range of c_j while c_j still cannot reach the gateway. Then c_j should be added to the listening subset of node c_i.

Adding such an extension to the protocol imposes an additional level of complexity in the implementation. So it may be used only if the obtained gain is convincing. We investigated such a mechanism in real WBAN deployments in our experiments. The results are discussed in Sec. 6.5.

6.4.5 Multi-Hop Data Forwarding

When there is no node disconnected from the gateway ($\Psi^t = \emptyset$), then every body node only transmits its own data. Otherwise, a routing mechanism should be exploited for multi-hop data propagation. As the network topology of the WBAN is always prone to change, any assumption about the position of the nodes and their connectivity is not reliable. Therefore, we use a gossip-based strategy [57], which does not rely on any routing structure and so is robust against frequent changes in the WBAN topology.

Nodes use a store-and-forward mechanism for relaying the data items from the nodes in the requested set. The node c_i may receive packets from the nodes that it is listening to. The node then maintains a cache of the data items that it has received and has to forward. At any time, a buffer of the last $\lceil T_{tx}(c_j)/T_s(c_j) \rceil$ received data items originated from the node c_j is stored in the data cache. The older data items from c_j will be removed from the cache once the newer data items arrive. Once the node c_j is taken out of the requested set Ψ^t, all its data items are deleted from the data cache of any node in the network, as it no longer needs data forwarding help. This provides a better chance for other data items in the data cache to be forwarded. Every sensor node is primarily responsible for propagation of its own sampled data. In addition, other data items that it may have received from the disconnected nodes will be forwarded.

Algorithm 5 shows the behavior of the body nodes in a network using ODLF, assuming a TDMA-based MAC layer with fixed slot assignment to the nodes in the WBAN. For every TDMA slot of the active part of the frame, a body node c_i first checks if the slot

Algorithm 5: Behavior of body sensor node c_i.

```
1  foreach active slot k do
2      if Owner(k) = c_i then                                   /* own tx slot */
3          TransmitPacket(TxPacket);
4      else if Owner(k) ∈ Ψ^{t-1} then                          /* in requested set */
5          RxPacket=ListenToSlot();
6          if RxPacket ≠ Nil then
7              foreach RxItem ∈ RxPacket do
8                  StoreInDataCache(RxItem);
9              end
10         end
11     else if Owner(k) = c_M then                              /* the gateway slot */
12         RxPacket=ListenToSlot();
13         if RxPacket ≠ Nil then
14             Ψ^t ⟵ RxPacket.Ψ − {c_i};
15         else
16             Ψ^t ⟵ Ψ^{t-1};
17         end
18     end
19 end
20 forall the node c_l ∈ C − {c_M} do
21     if c_l ∉ Ψ^t then
22         FlushDataCache(c_l);
23     end
24 end
25 TxPacket ⟵ {OwnDataItems} ∪ SelectFromDataCache();
```

is its own transmit slot (line 2). In this case, it transmits its data packet. Otherwise, it checks if the owner of the slot is a body node that is among the nodes in the requested set (line 4). If so, the node listens in the slot and stores the received packet, if any. But, if the slot is not for a node in the requested set, then node c_i listens to the channel in that slot only if the slot is owned by the gateway. The node then extracts the new requested set from the packet received from the gateway (line 14). Note that it excludes itself from the requested set, if node c_i itself is in the set. However, if no packet is received from the gateway, the node continues using the previously received requested set (line 16). After the active part of the TDMA frame, the node removes the data items in its cache whose source nodes are not in the requested set anymore. The last line of the algorithm prepares a data packet for transmission in the next TDMA frame. The node includes its own data sample as well as some items from its data cache. The protocol stack implementation for the ODLF mechanism is explained in the next section.

6.5 Experiments

We performed several experiments using wireless nodes deployed on a body to verify the functionality of the proposed mechanism and to compare it with other approaches. In this section, the experimental setup is explained and the analysis of the obtained result is presented.

6.5.1 Experimental Setup

TDMA is widely used for WBANs ([49, 54, 57, 65, 70]) as a suitable MAC layer, which outperforms other mechanisms such as CSMA-based protocols in many aspects. Long sleep periods and contention-free behavior of the TDMA strategy make the protocol energy efficient with maximal bandwidth utilization. Thanks to the small deployment area and the presence of a more powerful node (the gateway) playing a coordinating role, the synchronization between nodes in a WBAN is quite reliable. Moreover, the limited network size allows a very small duty cycle. We use a TDMA-based MAC layer in our protocol stack for intra-WBAN communication. The structure of frames and time slots are presented in Chapter 2. A fixed slot allocation is used to assign exclusive transmit slots to the body sensor nodes in the WBAN. Since the slots are unique, the protocol is contention free (provided that a synchronization mechanism is in place). Every node transmit a packet in its dedicated time slot at each round. Thus $T_{tx}(c_i) = T_{frame}$ for any node c_i in the WBAN. The gateway node listens to all nodes in every frame. Other nodes always listen to the gateway slot as well as the slots of any node that is requested by the protocol. Therefore, the minimum radio activity of a node in a frame consists of the transmission of its packets in its dedicated slot and listening to the gateway slot. More listening activity occurs only when invoked by the data routing mechanism.

Listening to the gateway slot in each round is crucial for the protocol. The gateway packets are also used as a beacon for synchronization purposes. Any node in each frame aligns its time frame with the gateway. If a node does not receive a packet from the gateway in some exceptional cases, to prevent loosing synchronization, it starts to listen to other nodes. However, this does not occur in normal circumstances, as the gateway transmission range is supposed to be high enough to cover all nodes in all postures.

We implemented the ODLF mechanism as well as the TDMA-based MAC layer and the gossip-based data forwarding mechanism, duscussed in Chapter 2, on the MyriaNed nodes for our experiments. The TDMA frame length is set to one second ($T_{frame} = 1$ sec.). We deployed 11 nodes (including the gateway) on different positions of the body, shown in Fig. 6.4. Node c_{10} is deployed on the back. A node connected to the computer is used to configure the WBAN and control the experiments. This node is not a part of the WBAN. During each experiment, body nodes log the radio activities and the content of their data cache in their flash memory at each TDMA round. After finishing each experiment, the logged data is downloaded from the nodes and is analyzed to extract performance metrics.

Several experiments of in total 30 hours have been done each lasting one hour with a particular configuration. This is done to observe the behavior of the protocol in vari-

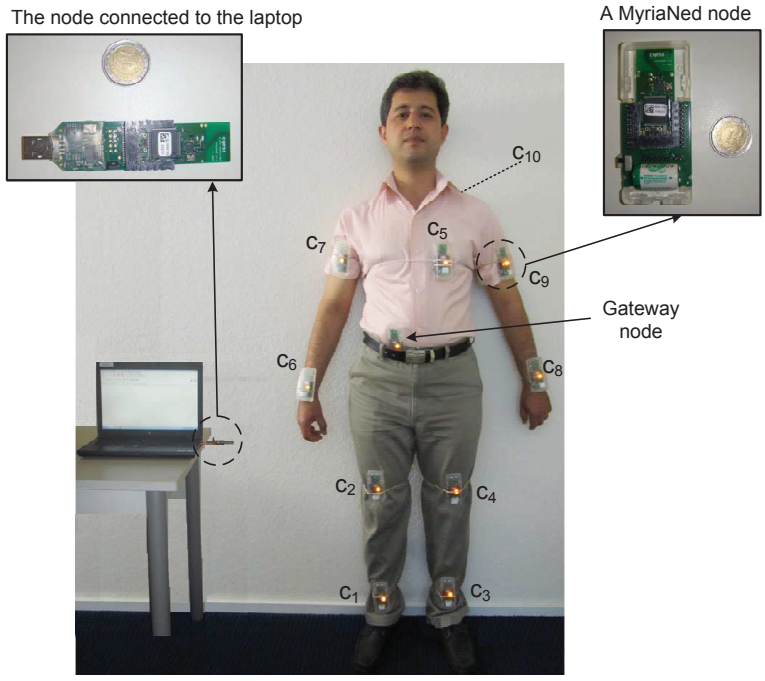


Figure 6.4: Position of the nodes deployed on the body in the experiments.

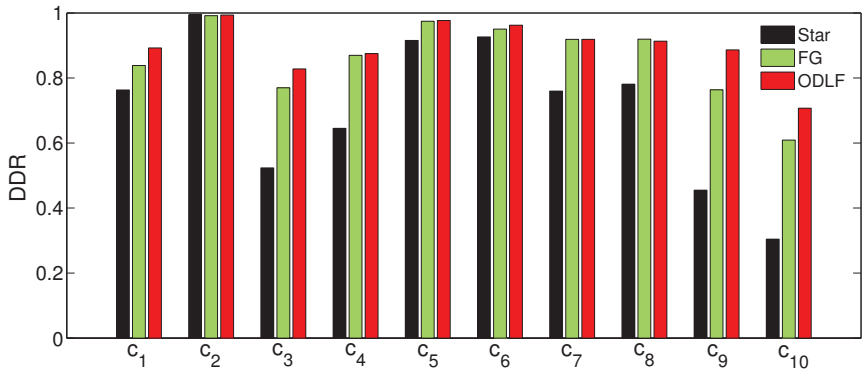


Figure 6.5: Average DDR of nodes to the gateway over all experiments exploiting different protocols.

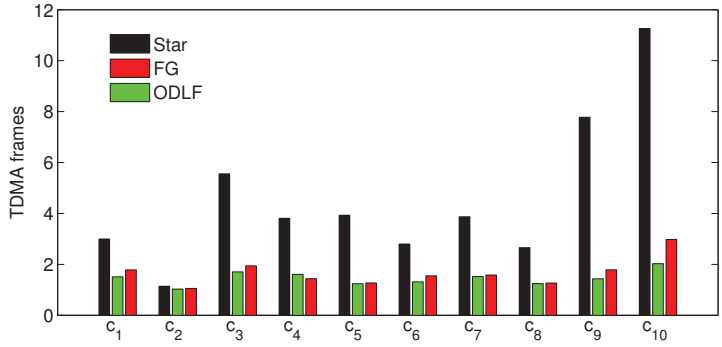
ous circumstances. The configurations include different transmit power levels, different sampling periods (T_s = 1, 3, or 5 sec.), and different posture patterns. To simplify the process of data analysis and presentation, we assume the same settings for all nodes during each setup. A combination of *walking*, *standing*, *sitting*, and *lying down* postures is set as the posture pattern for each experiment. Each experiment is repeated three times, with the same settings, using the three different data propagation mechanisms: the star approach, full gossip data forwarding as presented in [57] (FG), and the proposed on-demand listening and forwarding strategy (ODLF). Although the same posture patterns and similar movements are used for three experiments of a setup, the actual connectivity and link quality cannot be guaranteed to be exactly the same. However, according to our measurements, the average size of the requested set is very similar for the three experiments in each setup, which indicates a similar average connectivity to the gateway. All experiments were done in an indoor environment.

We calculate several metrics from the logged information in the flash memory of the nodes, including the gateway node. In particular, we calculate the achieved DDR to the gateway in each experiment, to evaluate the performance of each approach in delivering data items. In a star strategy, the data originating from node c_i is lost if the direct link from node c_i to the gateway fails for $T_s(c_i)/T_{frame}$ subsequent rounds. There is also a chance of data loss in the multi-hop protocols. Having no connection to any node can be a source of data loss, because none of the nodes in the WBAN can help in this situation. We also consider the length of burst data losses, as explained in Chapter 2.

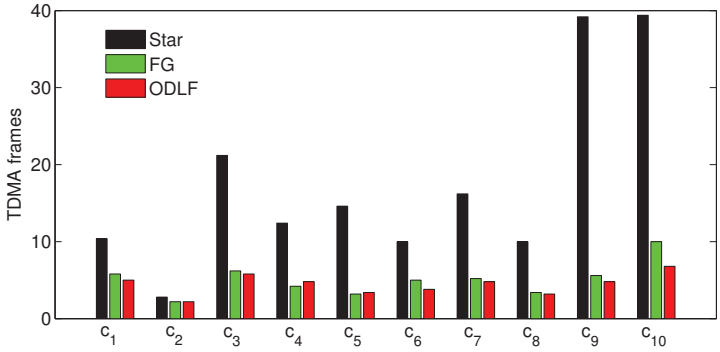
The energy consumption of sensor nodes for MyriaNed nodes is calculated by Eqn. 2.9 explained in Chapter 2. The value of parameter N_{listen} (the number of listening slots in a frame) in Eqn. 2.11 for calculating the listening energy consumption is 1, $1 + |\Psi^t|$, and M for the Star, ODLF, and FG mechanisms, respectively. Note that in the protocol stack of the experiments, body nodes listen to the gateway slot in every TDMA frame. This is the minimum listening activity of each node.

6.5.2 Data Analysis and Results

To get a general impression of the behavior of different protocols, the average results obtained from all performed experiments are presented. Fig. 6.5 shows the DDR values over all experiments, individually shown for each node and each protocol stack. This result shows that, taking all experiments with different setups into account, our ODLF protocol provides much better DDR than the Star approach, and almost the same or slightly better values than the FG protocol. The latter may seem counterintuitive. Below, we clarify this result. Considering the average result for all nodes, in these experiments, the ODLF mechanism gives a 27% improvement for DDR in comparison with the Star network. Note that for some nodes, like c_2, the DDR is always very good, as it can always reach the gateway, so there is almost no room for improvement by exploiting ODLF. In contrast, for some nodes with lower link quality to the gateway, the improvement is considerable. In one particular experiment, for instance, the DDR value in the star network is smaller than 20% for four nodes, whereas values of more than 75% are achieved for those nodes by the ODLF protocol.



(a) Average length of burst data loss over all experiments



(b) The upper bound length of 95% of bursts

Figure 6.6: Distribution of the length of burst data loss.

Fig. 6.6 gives some information about the distribution of the length of burst data losses ($\mathcal{BL}^t(c_i)$) during all experiments using different protocols. Fig. 6.6(a) depicts the average length over all experiments using each mechanism. Fig. 6.6(b) presents the smallest upper bound on the length of 95% of the data loss bursts. The main observation here is that in many cases during the experiments, continuous outage has happened for some nodes in the network. This could happen because of the node position in specific postures for a sustained period. Of course, some nodes do not have such a problem because they have a good link to the gateway. Nodes c_3, c_4, c_9, and c_{10} showed to be the most risky nodes in the experiments. Again, the obtained result using the ODLF approach is considerably better than the result for the star network and slightly better than for the FG approach.

Compared to FG, the ODLF protocol gives an average improvement of 4%, 12% and 41%, for DDR, average and maximum burst loss, respectively. The reason for such improvement is that in the FG mechanism, the nodes always listen to all other nodes.

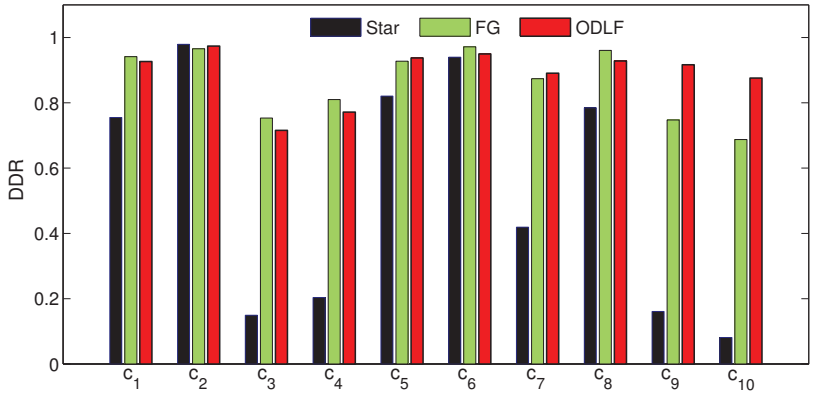


Figure 6.7: DDR of different body sensor nodes to the gateway in an experiment with low quality links.

Thus, many packets may be received, which should be stored in the data cache and forwarded. Because there are more data items in the data cache, each data item has less chance to be selected from the data cache for transmission in a round. Therefore, there is a higher probability that a data item is overwritten by the next version before being transmitted. In contrast, in ODLF, a node only listens to the nodes in the requested set. This means fewer data items in the cache and a bigger chance for the data items from the nodes in the requested set to be forwarded.

The ODLF mechanism achieves improvements at the expense of more energy consumption than the star strategy whenever data forwarding is necessary. The average energy consumption of nodes per frame over all experiments is 82μJ, 284μJ, and 138μJ for star, FG, and ODLF protocols, respectively. Comparing ODLF with FG, we observe a big gain in the energy saving with the same or better QoS. Comparing our protocol with the star network shows that we may spend more energy to make WBAN application feasible and achieve much better QoS. As the amount of radio activity in a frame is always fixed in star and FG approaches, the energy consumption is almost the same in all experiments (minor variation is caused by having different transmit power levels). In contrast, the connectivity to the gateway plays a major role in energy consumption using the ODLF mechanism. In a network with poor links, the requested set is bigger and nodes should listen in more slots. In a WBAN where nodes can always reach the gateway, the requested set is almost empty and the energy consumption is as low as in the star network.

To further investigate the behavior of the protocol in different network situations, we present more details about the result of two experimental setups, one with the best connectivity (all nodes use the highest transmit power level) and another with the worst connectivity (lowest transmit power level). In the former network, the requested set is almost empty. The average energy consumption per frame is 86μJ, which is quite close to

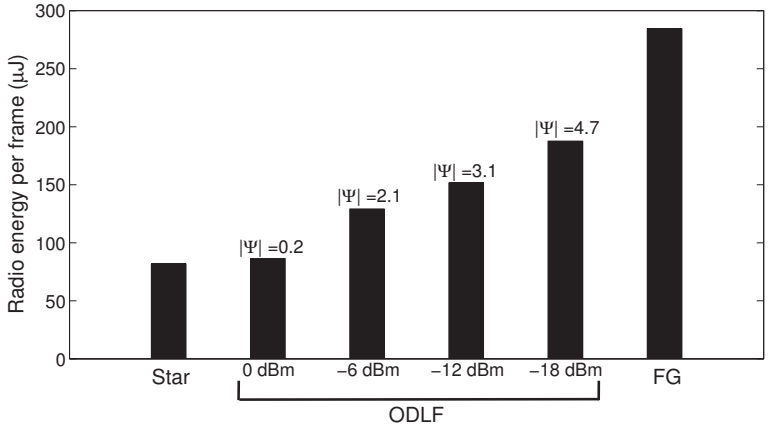


Figure 6.8: Average energy consumption using ODLF protocol in experiments with different link quality (different transmit power levels of the body nodes), and the consumption in star and FG protocols. Note that in star and FG protocols, the listening energy consumption does not depends on the quality of links.

the one of the star network (82μJ for this experiment). The minor energy consumption overhead is for some rare cases during the experiment that some nodes (in particular c_{10}) cannot reach the gateway for short periods in some postures such as lying down. On average, an almost full packet delivery (99.8%) was provided in this experiment by using ODLF whereas there is slightly more data loss in the star network (DDR of 97.8% on average). The achieved DDR for node c_{10}, for instance, is 99.4% and 85.7% using the ODLF and star mechanisms, respectively. This means that in a WBAN with proper links to the gateway, our ODLF mechanism spends almost the same energy as the star strategy. However, ODLF guarantees data forwarding if the gateway dropped out of reach of a node during network operation, giving a better DDR. Note that such a good connectivity to the gateway is not feasible in many applications of WBANs. Specifically designed wireless body sensor nodes with very small size (short antenna) and nodes using energy scavenging may have very limited transmit range.

For the experiment with low network connectivity, the achieved DDR using ODLF is substantially improved compared to the DDR provided by the star topology. Fig. 6.7 depicts the DDR values in this experiment using different protocols. Nodes c_3, c_4, c_9, and c_{10} showed to have a very poor link to the gateway in this experiment. The achieved DDR for these nodes using ODLF is almost three times better than in the star network. The average length of burst data loss over all nodes is around 9 frames in the star network whereas its value is around 1.3 frames using ODLF. Of course this result has been achieved by spending more energy. However, the energy consumption using ODLF (187μJ) is very lower than the energy consumption in the FG approach. This is because the average size of the requested set was around 5 in this experiment. This experiment

Table 6.2: Result of subsetting the requested set.

performance metric	*base ODLF protocol*	*ODLF with extension*
DDR	94%	86%
Avg. burst loss	1.3 sec.	1.7 sec.
Max. burst loss	6 sec.	8.4 sec.
Energy per frame	187 μJ	144 μJ

confirms that for a network of which the nodes have limited radio range, for instance because of ultra-low power RF radio, ODLF allows the WBAN to work with a sufficient QoS level.

It is shown by the experiments that our hybrid protocol improves the performance metrics. According to the current connectivity status of the WBAN, it turns to a star topology or to multi-hop gossip-based data forwarding. Fig. 6.8 shows the average energy consumption of the ODLF protocol in several experiments with different transmit power levels used by the body nodes (in each experiment, all body nodes use the same transmit power levels). Using higher transmit power leads to better connectivity of body nodes to the gateway. The average size of the requested set ($|\Psi|$) in each experiment using ODLF is also shown as an indication of the connectivity to the gateway. The figure also presents the radio energy consumption using the star and FG protocols as two extremes.

6.5.3 Further Power Saving

As discussed in Sec. 6.4.4, it may be possible to further reduce energy consumption of nodes by maintaining a subset of the requested set from which the node can hear well. Thus, node c_i only listens to such a subset in each round instead of listening to all the nodes in the requested set. Such a subset should be carefully updated. We performed an experiment to investigate the functionality and performance gain of such an extension. To implement it, we use time intervals for listening to the nodes in the requested set. So c_i listens to c_j once per interval. When node c_i detects that it has not received anything from c_j in the past three frames, it starts to increase its interval for c_j by one up to a given maximum point (10 in our experiment). If a packet is received from c_j in a round, the size of the interval drops to one to allow a maximal listening to c_j. Note that this is done while c_j is in the requested set. Such strategy (additive increase and sudden drop) is exploited to minimize the amount of data loss.

We conducted two experiments with the same posture patterns and node settings, one running the base protocol and another using this extension. Table 6.2 presents the result obtained from these two experiments. The average energy consumption saving is 23% at the expense of 10% decrease of the DDR in comparison with the base protocol. It means that during some periods of the experiment, not providing forwarding help for some nodes in the requested set leads to data loss. The result shows a trade-off between achieved QoS and energy consumption. To make the final decision about using such an extension, the objectives of the target application should be taken into consideration.

6.6 Summary

This chapter focused on the communication in gateway-based WBANs in which all body nodes send their data to a gateway node on the body. The goal here is to provide reliable and efficient data delivery from body nodes to the gateway, considering real challenges of these networks such as high mobility, and low and time-variant quality of links. The chapter proposes an on-demand listening and data forwarding (ODLF) mechanism that adapts the data propagation strategy according to the quality of the links to the gateway. Nodes with a poor link receive data forwarding help through multi-hop gossiping. In a network with sufficiently good connectivity to the gateway, the protocol works like a star network. When some nodes cannot reach the gateway properly, other body nodes provide data forwarding to deliver their data to the gateway. Several experiments are done to observe the behavior of the protocol in a real deployment. The results show a big gain in DDR in comparison with star networks, especially for networks with low quality links. Moreover, a considerable gain is observed in energy consumption in comparison with a full gossip strategy. Experiments with various configurations of nodes show that the ODLF mechanism improves DDR of the body nodes to the gateway, in particular in networks with low connectivity.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

Pervasive health monitoring is one of the target application domains for Wireless Sensor Networks (WSNs). The combination of Wireless Body Area Networks (WBANs) and ambient sensor networks can provide an environment for aging people or people with particular diseases to receive support and sufficient care while living independently. Such networks for pervasive health applications exhibit characterizing specifications that differentiate them from other WSN applications. Mobility of WBANs, heterogeneity and dynamics in the network, and stringent power consumption constraints are challenges that should be taken into consideration for designing an appropriate communication infrastructure for these applications. Common communication protocols and standards for WSNs may be inefficient for specific applications and need to be adapted or optimized to be exploited for this particular application domain. This needs to be done towards giving the network the potential to fulfill the application requirements.

This thesis aims to provide dedicated communication mechanisms and services for health applications. A Medium Access Control (MAC) mechanism for supporting WBAN mobility as well as a data prioritization mechanism for data dissemination are developed for communications in an integrated ambient and WBAN network. Moreover, an adaptive protocol is proposed for communication of the body sensor nodes within a WBAN.

All proposed mechanisms in this thesis are evaluated through computer simulations as well as real-world experiments. All protocols are implemented in the MiXiM [47] framework on top of the OMNeT++ [5] network simulator and several simulations are performed with different setups to observe the behavior of the network. Relevant existing protocols are also simulated to enable a comparison with respect to performance. Moreover, in this thesis, we develop a configurable mobility model, MoBAN, for simulating intra- and extra-WBAN communication solutions. It models different body postures and individual node mobility. It also supports modeling spatial and temporal correlations of WBAN movements and posture patterns. MoBAN is implemented in the MiXiM frame-

work and is used for simulating the proposed mechanisms in this thesis. The MoBAN model is also available to the scientific community as it is integrated in the MiXiM 2.1 release.

All mechanisms are also implemented on MyriaNed [95] wireless nodes and several large-scale experiments are performed to further evaluate the performance of the protocols in real-world networks. We use a testbed of 76 wireless nodes with various setups. We deploy a static network on two floors of the Electrical Engineering department of the Eindhoven University of Technology (TU/e). Several people took part by wearing wireless nodes and by performing their normal behavior during office work. Real experiments prove the implementation feasibility of the proposed protocols by considering computation and communication limitations of real WSN nodes. Moreover, experiments provide more realistic results because simulation models cannot completely and perfectly mimic the real network behavior. Using a real network deployment ensures that the protocol works in a realistic setup.

This thesis proposes a mechanism for supporting WBAN mobility for TDMA-based MAC protocols using a flat network architecture. It exploits a hybrid schedule-based and contention-based approach to manage accessing the shared medium by different nodes in the network. In this protocol, the body nodes directly communicate with the static nodes in their neighborhood. Moreover, a listening scheduling mechanism is proposed to optimize energy consumption of the nodes by reducing idle listening to static nodes to WBAN nodes. Nodes in the ambient network estimate the hop distance of the WBANs in the network to schedule their listening activities. This leads to a substantial decrease of unnecessary listening activities of the nodes (around 70% reduction in the experiments and simulations) that considerably saves energy, without negative influence on the other performance metrics.

There is a high spatial and temporal heterogeneity in typical health application in terms of sensing and quality-of-service (QoS) requirements. For disseminating the sensed data, a data prioritization mechanism is of great importance to provide proper services to the data items sampled by different nodes in the network. This thesis proposes a dynamic data prioritization mechanism to support heterogeneity in the QoS requirements. It also supports varying requirements over time enabling autonomous multi-scenario applications. This mechanism is integrated in a gossip-based data propagation mechanism and realizes robust, fair, and reliable data delivery. The experimental results show that using the prioritization mechanism, the data items with more stringent QoS demands receive better service at the cost of less but sufficient service for less demanding data.

In some applications, body sensor nodes need to communicate through a gateway on the body. Considering the specifications of the WBANs such as mobility and the quality of wireless links, such communication within the WBAN is also very challenging. We propose an on-demand listening and data forwarding mechanism for intra-WBAN communication that automatically adapts the network topology to the connectivity status of the network from a one-hop star topology to a desired level of multi-hop gossip-based data propagation. The aim is to prevent long outage of nodes while optimizing energy consumption. Several experiments with deploying at least 12 MyriaNed wireless nodes on a human body are performed to evaluate the performance of the approach. Results

show that this mechanism makes a good trade-off between the energy consumption and data delivery ratio in comparison with fixed architectures.

7.2 Recommendations for Future Research

Exploiting WSNs for pervasive health monitoring is an ongoing topic under research and development. Although great research has been conducted in the field, there is still room for further investigation and research in efficiently utilizing current and future advances in micro-sensor wireless devices to improve the performance of the system. The general roadmap in this field is to 1) develop flexible, tiny, and wearable body sensor devices to make their usage more convenient for the target users, possibly using an energy scavenging technology; 2) make a global connected network of people, their doctors, and care centers. In this development, security of the network against malicious attacks and people's privacy are very important issues. Network robustness and reliability is also crucial because the network would be an important part of the healthcare system and a network crash can be very dangerous and costly. These are in general some areas that demand a lot of academic research.

There is also specific future work for the topics and mechanisms that have been studied in this thesis. The following recommends some future work for each part of the thesis.

- In Chapter 3, we present the MoBAN mobility model for WBANs that encompass posture changes and individual mobility. Posture changes can substantially change the quality of the wireless links between different nodes and their dynamics. Therefore, it is expected that posture is taken into account in the exploited radio model for simulation of WBANs. To the best of our knowledge, there is currently no dedicated radio model for WBANs that can properly model the radio link characteristics in the WBANs according to the posture of the body. However, considering the highly dynamic topology and the position of the body in the wave propagation path and the fading and shadowing effects caused by the human body, modeling the radio demands for substantial research. A radio model based on some empirical link data integrated with the MoBAN mobility model can provide a more precise model for simulating protocols for WBANs.

- In Chapter 4, body nodes use a contention-based medium access to send their data to the static nodes. It is done in the specified part of the TDMA frames, called MCS. For applications with a high number of WBANs, we use more MCS parts with a fixed MCS assignment to different WBANs. In such case, the static nodes need to listen to several MCS sections even if each MCS is being used only by one WBAN in the neighborhood. Nodes may exploit a distributed algorithm for dynamic MCS selection to optimize the number of MCS sections in a neighborhood. This can save considerable listening energy for static nodes.

- The data prioritization mechanism presented in Chapter 5 has been integrated with a gossip-based data dissemination [29] for performance evaluation. However, the

scope of the mechanism is not limited to only this type of data dissemination. Exploiting dynamic QoS-based data prioritization on top of other routing techniques (e.g., tree-based routing) and observing its performance is considered as future work on this topic. This mechanism can be further equipped with a load balancing algorithm to enhance the achieved quality of service.

- A WBAN can have a highly heterogeneous setup including sensor nodes whose data sampling and transmission rates substantially deviate from each other. Some body sensors need a high data propagation bandwidth (e.g., ECG or EEG sensors with a few hundred samples per second) while some others may generate very sporadic data (e.g., a SpO2 sensor with a few measurements per day). Supporting such diversity in the WBAN communication protocol can lead to more efficient data delivery and energy optimization.

- In Sec. 6.4, we use certain techniques for estimating the quality of wireless links between different nodes in the WABN and the gateway. It is then used for detecting disconnections from the gateway to trigger the on-demand listening and data forwarding (ODLF) mechanism. There are several link quality estimators that are presented in the literature for WSNs. Links in WBANs behave differently than in other WSNs, e.g., considering the dynamism in the network. Therefore, studying the performance of the existing link quality estimation methods for detecting and following connection changes in WBANs is worth considering to select the best approach. Using experimental data for such investigation is greatly recommended.

Bibliography

[1] INET Framework website. http://inet.omnetpp.org.

[2] INETMANET Framework website. http://github.com/inetmanet/inetmanet.

[3] Network Simulator-NS2/3 website. http://www.isi.edu/nsnam/ns/.

[4] NORDIC Semiconductor. nRF24L01 Radio Transceiver, 2007.

[5] OMNeT++ website. http://www.omnetpp.org.

[6] Texas instruments cc2420 radio transceiver, http://focus.ti.com/docs/prod/folders/print/cc2420.html, 2010.

[7] Veins - Vehicles in Network Simulation website. http://veins.car2x.org/.

[8] IEEE standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements. - part 15.1: Wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPANs). In *IEEE Std 802.15.1-2005 (Revision of IEEE Std 802.15.1-2002)*, pages 1–580, 2005.

[9] IEEE standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements. - part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. In *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, pages C1–1184, 2007.

[10] IEEE standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements. - part 15.4: Low-rate wireless personal area networks (LR-WPANs). In *IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006)*, pages 1–314, 2011.

[11] IEEE standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements. - part 15.6: Wireless body area networks (WBANs). In *IEEE Std 802.15.6-2012*, pages 1–271, 2012.

[12] M. Ali, T. Suleman, and Z.A. Uzmi. MMAC: a mobility-adaptive, collision-free MAC protocol for wireless sensor networks. In *Proc. 24th IEEE international Performance, Computing, and Communications Conference (IPCCC)*, pages 401–407. IEEE, 2005.

[13] F. Assegei. Decentralized frame synchronization of a TDMA-based wireless sensor network. Master's thesis, Eindhoven University of Technology, the Netherlands, 2008.

[14] N. Baccour, A. Koubaa, L. Mottola, M.A. Zuniga, H. Youssef, C.A. Boano, and M. Alves. Radio link quality estimation in wireless sensor networks: A survey. *ACM Transations on Sensor Networks (TOSN)*, 8(4), 2012.

[15] M. Blagojevi, M. Nabi, M.C.W. Geilen, T. Basten, T. Hendriks, and M. Steine. A probabilistic acknowledgment mechanism for wireless sensor networks. In *Proc. 6th IEEE International Conference on Networking, Architecture, and Storage (NAS)*, pages 28–30. IEEE, 2011.

[16] M. Blagojevic, M.C.W. Geilen, T. Basten, and T. Hendriks. Fast sink placement for gossip-based wireless sensor networks. In *Proc. 31st IEEE International Performance Computing and Communications Conference (IPCCC)*, pages 110–119. IEEE, 2012.

[17] M. Buettner, , G.V. Yee, E. Anderson, and R. Han. X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks. In *Proc. Conference on Embedded Networked Sensor Systems (SenSys)*, pages 307–320. ACM, 2006.

[18] J. Cai, S. Cheng, and C. Huang. MAC channel model for WBAN. Tech. report, 15-09-0562-00-0006, IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs), 2009.

[19] S. Cai, Y. Liu, and W. Gong. Analysis of an AIMD based collision avoidance protocol in wireless data networks. In *Proc. 42nd IEEE Conference on Decision and Control*, pages 104–109. IEEE, 2003.

[20] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing (WCMC): Special Issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2(5):483–502, 2002.

[21] J. Chen, M. Zhou, D. Li, and T. Sun. A priority based dynamic adaptive routing protocol for wireless sensor networks. In *Proc. International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*, pages 160–164. IEEE, 2008.

[22] T.V. Dam and K. Langendoen. An adaptive energy-efficient MAC protocol for wireless sensor networks. In *Proc. Conference on Embedded Networked Sensor Systems (SenSys)*, pages 171–180. ACM, 2003.

[23] W.J. Dixon and F.J. Massey. *Introduction to statistical analysis*. McGraw-Hill, 1983.

[24] A. El-Hoiydi and J.-D. Decotignie. WiseMAC: an ultra low power MAC protocol for the downlink of infrastructure wireless sensor networks. In *Proc. 9th International Symposium on Computers and Communications (ISCC)*, pages 244–251. IEEE, 2004.

[25] A. Ephremides and O.A. Mowafi. Analysis of a hybrid access scheme for buffered users-probabilistic time division. *IEEE Transactions on Software Engineering (TSE)*, SE-8(1):52–61, 1982.

[26] S.C. Ergen and P. Varaiya. PEDAMACS: Power efficient and delay aware medium access protocol for sensor networks. *IEEE Transactions on Mobile Computing (TMC)*, 5(7):920–930, 2006.

[27] T. Falck, H. Baldus, J. Espina, and K. Klabunde. Plug 'n play simplicity for wireless medical body sensors. *Mobile Networks and Applications*, 12(2-3):143–153, 2007.

[28] R. Fonseca, O. Gnawali, K. Jamieson, and P. Levis. Four bit wireless link estimation. In *Proc. Sixth Workshop on Hot Topics in Networks (HotNets VI)*, 2007.

[29] D. Gavidia and M. van Steen. A probabilistic replication and storage scheme for large wireless networks of small devices. In *Proc. 5th IEEE International Conference Mobile and Ad Hoc Sensor Systems (MASS)*, pages 469–476. IEEE, 2008.

[30] S.M. George, W. Zhou, H. Chenji, M. Won, Y.O. Lee, A. Pazarloglou, R. Stoleru, and P. Barooah. DistressNet: a wireless adhoc and sensor network architecture for situation management in disaster response. *IEEE Communications Magazine*, 48(3):128–136, 2010.

[31] S.K.S. Gupta, S. Lalwani, Y. Prakash, E. Elsharawy, and L. Schwiebert. Towards a propagation model for wireless biomedical applications. In *Proc. IEEE International Conference on Communications (ICC)*, pages 1993–1997. IEEE, 2003.

[32] J.-H. Hauer, V. Handziski, and A. Wolisz. Experimental study of the impact of WLAN interference on IEEE 802.15.4 body area networks. In *Proc. 6th European Conference on Wireless Sensor Networks (EWSN)*, pages 17–32. Springer-Verlag, 2009.

[33] T. He, S. Krishnamurthy, L. Luo, T. Yan, L. Gu, R. Stoleru, G. Zhou, Q. Cao, P. Vicaire, J.A. Stankovic, T.F. Abdelzaher, J. Hui, and B. Krogh. VigilNet: an integrated sensor network system for energy-effcient surveillance. *ACM Transactions on Sensor Networks (TOSN)*, 2(1):1–38, 2006.

[34] X. Hong, M. Gerla, G. Pei, and C.-C. Chiang. A group mobility model for ad hoc wireless networks. In *Proc. 2nd ACM Interntional Conference on Modeling, analysis and Simulation of Wireless and Mobile systems (MSWiM)*, pages 53–60. ACM, 1999.

[35] E. Hyytiä and J. Virtamo. Random waypoint mobility model in cellular networks. *Wireless Networks*, 13(2):177–188, 2007.

[36] F. Ingelrest, G. Barrenetxea, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange. SensorScope: application-specific sensor network for environmental monitoring. *ACM Transactions on Sensor Networks (TOSN)*, 6(2):17:1–17:32, 2010.

[37] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva. Directed diffusion for wireless sensor networking. *IEEE/ACM Transactions on Networking (TON)*, 11(1):2–16, 2003.

[38] A. Jhumka and S. Kulkarni. On the design of mobility-tolerant TDMA-based media access control (MAC) protocol for mobile sensor networks. In *Proc. 4th International Conference on Distributed Computing and Internet Technology (ICDCIT)*, pages 42–53. Springer-Verlag, 2007.

[39] A.J. Johansson. Wave-propagation from medical implants-influence of body shape on radiation pattern. In *Proc. 24th Conference of the Biomedical Engineering Society*, volume 2, pages 1409–1410. IEEE, 2002.

[40] D.B. Johnson, D.A. Maltz, and J. Broch. DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks. In *In Ad Hoc Networking, Chapter 5*, pages 139–172. Addison-Wesley, 2001.

[41] E. Jovanov, A. Milenkovic, C. Otto, and P. Groen. A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation. *Network*, 2(6), 2005.

[42] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.S. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet. *ACM SIGOPS Operating Systems Review*, 36(5):96–107, 2002.

[43] B.A. Kadrovach and G.B. Lamont. Design and analysis of swarm-based sensor systems. In *Proc. 44th IEEE Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 487–490. IEEE, 2001.

[44] B.A. Kadrovach and G.B. Lamont. A particle swarm model for swarm-based networked sensor systems. In *Proc. ACM Symposium on Applied Computing (SAC)*, pages 918–924. ACM, 2002.

[45] A. Khadivi and M. Hasler. Fire detection and localization using wireless sensor networks. *Sensor Applications, Experimentation, and Logistics*, 29:16–26, 2010.

[46] S. Kim, S. Lee, H.-J. Ju, D. Ko, and S. An. Priority-based hybrid routing in wireless sensor networks. In *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6. IEEE, 2010.

[47] A. Kopke, M. Swigulski, K. Wessel, D. Willkomm, P.T. Llein Haneveld, T.E.V. Parker, O.W. Visser, H.S. Lichte, and S. Valentin. Simulating wireless and mobile networks in OMNeT++ - the MiXiM vision. In *Proc. International Conference on Simulation Tools and Techniques for Communications, Networks and Systems and Workshops (SIMUTools)*. ICST, 2008.

[48] K. Langendoen and A. Meier. Analyzing MAC protocols for low data-rate applications. *ACM Transactions on Sensor Networks (TOSN)*, 7(1):10:1–10:34, 2010.

[49] B. Latr, B. Braem, C. Blondia, I. Moerman, E. Reusens, W. Joseph, and P. Demeester. A low-delay protocol for multihop wireless body area networks. In *Proc. 4th International Conference on Mobile and Ubiquitous Systems (MobiQuitous)*, pages 1–8. IEEE, 2007.

[50] K. Lee, S. Hong, S.J. Kim, I. Rhee, and S. Chong. SLAW: A new mobility model for human walks. In *Proc. Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pages 855–863. IEEE, 2009.

[51] B. Liang and Z.J. Haas. Predictive distance-based mobility management for PCS networks. In *Proc. Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pages 1377–1384. IEEE, 1999.

[52] C. Liu and E. Modiano. On the performance of additive increase multiplicative decrease AIMD protocols in hybrid space-terrestrial networks. *Computer Networks and ISDN Systems*, 47(5):661–678, 2005.

[53] Y. Liu and W.K.G. Seah. A scalable priority-based multi-path routing protocol for wireless sensor networks. *International Journal of Wireless Information Networks*, 12(1):23–33, 2005.

[54] S.J. Marinkovic, E.M. Popovici, C. Spagnol, S. Faul, and W.P. Marnane. Energy-efficient low duty cycle MAC protocol for wireless body area networks. *IEEE Transactions on Information Technology in Biomedicine (TITB)*, 13(6):915–925, 2009.

[55] A. Mei and J. Stefa. SWIM: A simple model to generate small mobile worlds. In *Proc. Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pages 2106–2113. IEEE, 2009.

[56] P. Misra, N. Ahmed, and S. Jha. An empirical study of asymmetry in low-power wireless links. *IEEE Communications Magazine*, 50(7):137–146, 2012.

[57] M. Nabi, T. Basten, M.C.W. Geilen, M. Blagojevic, and T. Hendriks. A robust protocol stack for multi-hop wireless body area networks with transmit power adaptation. In *Proc. 5th International Conference on Body Area Networks (BodyNets)*. ICST, 2010.

[58] M. Nabi, M. Blagojevic, M.C.W. Geilen, and T. Basten. Dynamic data prioritization for quality-of-service differentiation in heterogeneous wireless sensor networks. In *Proc. 8th annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pages 296–304. IEEE, 2011.

[59] M. Nabi, M. Blagojevic, M.C.W. Geilen, T. Basten, and T. Hendriks. MCMAC: An optimized medium access control protocol for mobile clusters in wireless sensor networks. In *Proc. 7th annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pages 28–36. IEEE, 2010.

[60] M. Nabi, M.C.W. Geilen, and T. Basten. MoBAN: A configurable mobility model for wireless body area networks. In *Proc. 4th International Conference on Simulation Tools and Techniques (SIMUTools)*. ICST, 2011.

[61] M. Nabi, M.C.W. Geilen, and T. Basten. Demonstrating on-demand listening and data forwarding in wireless body area networks. In *9th annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), Demo Abstract*. IEEE, 2012.

[62] M. Nabi, M.C.W. Geilen, and T. Basten. On-Demand data forwarding for automatic adaptation of data propagation in WBANs. In *Proc. 9th annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pages 250–258. IEEE, 2012.

[63] A. Natarajan, B. de Silva, K.-K. Yap, and M. Motani. To hop or not to hop: Network architecture for body sensor networks. In *Proc. 6th annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pages 682–690. IEEE, 2009.

[64] A. Natarajan, M. Motani, B. de Silva, K.-K. Yap, and K.C. Chua. Investigating network architectures for body sensor networks. In *Proc. 1st ACM SIGMOBILE International workshop on Systems and Networking Support for Healthcare and Assisted Living Environments (HealthNet)*, pages 19–24. ACM, 2007.

[65] C. Otto, A. Milenkovic, C. Sanders, and E. Jovanov. System architecture of a wireless body area sensor network for ubiquitous health monitoring. *Journal of Mobile Multimedia*, 1(4):307–326, 2005.

[66] V. Pareto. Piccola biblioteca scientifica. *Manual of Political Economy*, pages 795–825, 1906. Translated into English by Ann S. Schwier (1971).

[67] Y. Peng, Z. Li, D. Qiao, and W. Zhang. Delay-bounded MAC with minimal idle listening for sensor networks. In *Proc. Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pages 1314–1322. IEEE, 2011.

[68] C.E. Perkins and E.M. Royer. Ad hoc on-demand distance vector routing. In *Proc. 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, pages 90–100. IEEE, 1999.

[69] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *Proc. ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pages 95–107. ACM, 2004.

[70] K.S. Prabh and J.-H. Hauer. Opportunistic packet scheduling in body area networks. In *Proc. 8th European Conference on Wireless Sensor Networks (EWSN)*, pages 114–129. Springer-Verlag, 2011.

[71] M. Quwaider and S. Biswas. On-body packet routing algorithms for body sensor networks. In *Proc. International Conference on Networks and Communications (NetCoM)*, pages 171–177. IEEE, 2009.

[72] M. Quwaider and S. Biswas. Probabilistic routing in on-body sensor networks with postural disconnections. In *Proc. 7th ACM International Symposium on Mobility Management and Wireless Access (MobiWAC)*, pages 149–158. ACM, 2009.

[73] V. Rajendran, K. Obraczka, and J.J. Garcia-Luna-Aceves. Energy-efficient collision-free medium access control for wireless sensor networks. In *Proc. Conference on Embedded Networked Sensor Systems (SenSys)*, pages 181–192. ACM, 2003.

[74] T.S. Rappaport. *Wireless Communication Principles and Practice*. Prentice Hall, Reading, MA, 2002.

[75] H. Ren and M.Q.-H. Meng. Understanding the mobility model of wireless body sensor networks. In *Proc. IEEE International Conference on Information Acquisition*, pages 306–310. IEEE, 2006.

[76] J. Ren, G. Wu, and L. Yao. A sensitive data aggregation scheme for body sensor networks based on data hiding. *Personal and Ubiquitous Computing*, 2012.

[77] E. Reusens, W. Joseph, B. Latre, B. Braem, G. Vermeeren, E. Tanghe, L. Martens, I. Moerman, and C. Blondia. Characterization of on-body communication channel and energy efficient topology design for wireless body area networks. *IEEE Transactions on Information Technology in Biomedicine (TITB)*, 13(6):933–945, 2009.

[78] I. Rhee, A. Warrier, M. Aia, J. Min, and M.L. Sichitiu. Z-MAC: a hybrid MAC for wireless sensor networks. *IEEE/ACM Transactions on Networking (TON)*, 16(3):511–524, 2008.

[79] I. Rhee, A. Warrier, J. Min, and L. Xu. DRAND: Distributed randomized TDMA scheduling for wireless ad hoc networks. *IEEE Transactions on Mobile Computing (TMC)*, 8(10):1384–1396, 2009.

[80] J.A. Lpez Riquelmea, F. Sotoa, J. Suardaza, P. Sncheza, A. Iborraa, and J.A. Verab. Wireless sensor networks for precision horticulture in southern spain. *Computers and Electronics in Agriculture*, 68(1):25–35, 2009.

[81] L. Roelens, S. van den Bulcke, W. Joseph, G. Vermeeren, and L. Martens. Path loss model for wireless narrowband communication above flat phantom. *IEE Electronics Letters*, 42(1):10–11, 2006.

[82] E. Romero, R. Warrington, and M.R. Neuman. Energy scavenging sources for biomedical sensors. *Physiological Measurement*, 30(9), 2009.

[83] R.R. Roy. *Handbook of Mobile Ad Hoc Networks for Mobility Models*. Springer, 2011.

[84] E.M. Royer, P.M. Melliar-Smith, and L.E. Moser. An analysis of the optimum node density for ad hoc mobile networks. In *Proc. IEEE International Conference on Communications (ICC)*, pages 857–861. IEEE, 2001.

[85] D. Sagan. RF integrated circuits for medical applications: Meeting the challenge of ultra low power communication. In *Ultra-Low-Power Communications Division*. Zarlink Semiconductor, 2005.

[86] M. Sanchez and P. Manzoni. ANEJOS: a java based simulator for ad hoc networks. *Future Generation Computer Systems*, 17(5):573–583, 2001.

[87] L. Sang, A. Arora, and H. Zhang. On link asymmetry and one-way estimation in wireless sensor networks. *ACM Transations on Sensor Networks (TOSN)*, 6(2), 2010.

[88] A. Sani. *Modelling and Characterisation of Antennas and Propagation for Body-Centric Wireless Communication*. PhD thesis, Queen Mary University of London, London, United Kingdom, 2010.

[89] M. Steine, M.C.W. Geilen, and T. Basten. Distributed maintenance of minimum-cost path information in wireless sensor networks. In *Proc. 6th ACM International Workshop on Performance Monitoring, Measurement and Evaluation of Heterogeneous Wireless and Wired Networks (PM2HW2N)*, pages 25–32. ACM, 2011.

[90] M. Steine, C.V. Ngo, R.S. Oliver, M.C.W. Geilen, T. Basten, G. Fohler, and J. Decotignie. Proactive reconfiguration of wireless sensor networks. In *Proc. 14th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, pages 31–39. ACM, 2011.

[91] M. Stemm and R.H. Katz. Measuring and reducing energy consumption of network interfaces in hand-held devices. *IEICE Transactions on Communications*, E80-B(8):1125–1131, 1997.

[92] E. Strmmer, M. Hillukkala, and A. Ylisaukkooja. Ultra-low power sensors with near field communication for mobile applications. In *Proc. International Conference on Wireless Sensor and Actor Networks (WSAN)*, pages 131–142. Springer, 2007.

[93] R. Tjoa, K.L. Chee, P.K. Sivaprasad, S.V. Rao, and J.G. Lim. Clock drift reduction for relative time slot TDMA-based sensor networks. In *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, volume 2, pages 1042–1047.

[94] F. van der Wateren. MyriaCore implementation details, the inside of MyriaCore and gMac. Technical report, Chess Company, the Netherlands, 2010.

[95] F. van der Wateren. Myrianed development guidelines, the art of developing WSN applications with MyriaNed. Technical report, Chess Company, the Netherlands, 2011.

[96] L. van Hoesel and P. Havinga. A lightweight medium access protocol (LMAC) for wireless sensor networks: Reducing preamble transmissions and transceiver state switches. In *Proc. International Workshop on Networked Sensing Systems (INSS)*, pages 205–208. Society of Instrument and Control Engineers (SICE), 2004.

[97] L. van Hoesel and P. Havinga. Collision-free time slot reuse in multi-hop wireless sensor networks. In *Proc. International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, pages 101–107. IEEE, 2005.

[98] L. van Hoesel and P. Havinga. Ideas on node mobility support in schedule-based medium access. In *Proc. International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, pages 539–544. IEEE, 2008.

[99] M. Welsh. Codeblue: Wireless sensors for medical care. [online]. available: http://fiji.eecs.harvard.edu/codeblue.

[100] A. Wood, J.A. Stankovic, G. Virone, L. Selavo, Z. He, Q. Cao, T. Doan, Y. Wu, L. Fang, and R. Stoleru. Context-aware wireless sensor networks for assisted living and residential monitoring. *Journal of NeuroEngineering and Rehabilitation*, 22(4):26–33, 2008.

[101] Fact sheet 999 World Health Organization (WHO). Epilepsy. `http://www.who.int/mediacentre/factsheets/fs999/en/`.

[102] G.-Z. Yang. Ubimon - ubiquitous monitoring environment for wearable and implantable sensors. [online]. available: http://www.doc.ic.ac.uk/vip/ubimon/research/index.html.

[103] G.-Z. Yang, editor. *Body Sensor Networks*. Springer-Verlag, 2006.

[104] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient MAC protocol for wireless sensor networks. In *Proc. Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pages 1567–1576. IEEE, 2002.

[105] W. Ye, F. Silva, and J. Heidemann. Ultra-low duty cycle MAC with scheduled channel polling. In *Proc. Conference on Embedded Networked Sensor Systems (SenSys)*, pages 321–334. ACM, 2006.

[106] T. Zasowski, F. Althaus, M. Stager, A. Wittneben, and G. Troster. UWB for noninvasive wireless body area networks: Channel measurements and results. In *Proc. Conference on Ultra Wideband Systems and Technologies (UWBST)*, pages 285–289. IEE, 2003.

[107] C. Zhao and M.L. Sichitiu. N-Body: Social based mobility model for wireless ad hoc network research. In *Proc. 7th annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pages 1–9. IEEE, 2010.

[108] Y. Zhao, A. Sani, Y. Hao, S. Lee, and G.-Z. Yang. A subject-specific radio propagation study in wireless body area networks. In *IEEE Conf. Loughborough Antennas and Propagation (LAPC)*, pages 80–83. IEEE, 2009.

[109] M.M. Zonoozi and P. Dassanayake. User mobility modeling and characterization of mobility patterns. *IEEE Journal on Selected Areas in Communications*, 15(7):1239–1252, 1977.

[110] A. Zurbuchen, A. Pfenniger, A. Stahel, C.T. Stoeck, S. Vandenberghe, V.M. Koch, and R. Vogel. Energy harvesting from the beating heart by a mass imbalance oscillation generator. *Annals of Biomedical Engineering*, 41(1):131–141, 2013.

Curriculum Vitae

Majid Nabi was born in Iran, on August 23, 1979. After graduating from high school in 1997, he started studying computer engineering (hardware) at Isfahan University of Technology, Isfahan, Iran. After receiving his Bachelor degree in 2001, he started working at the Information and Communication Technology Institute (ICTI) in Isfahan University of Technology, as a digital system designer for four years till 2005. He received his Masters with honors in Computer Architecture from the University of Tehran, Iran in September 2007. Afterward, he worked as a university lecturer for one year at Azad university in Najafabad, Iran.

Since September 2008, he has been a Ph.D. student within the Electronic Systems group at the Electrical Engineering Department of Technische Universiteit Eindhoven (TU/e) in Eindhoven, the Netherlands. His research was funded by the Dutch innovation program Point-One under project ALwEN: Ambient Living with Embedded Networks. His research led among others to a number of publications and this thesis. He was also involved in developing and teaching courses such as Digital Signal Processing and Networked Embedded Systems in the Electronic Systems group.

Majid is currently a post-doctoral researcher, continuing his research and teaching activities in the Electronic Systems group at the Electrical Engineering Department of Technische Universiteit Eindhoven.

List of Publications

The following is the list of publications of Majid Nabi in the field of wireless sensor and ad-hoc networks, when affiliated by Technische Universiteit Eindhoven. A part of these publications is covered in this thesis.

Publications covered in the thesis

- M. Nabi, M. Geilen, T. Basten, and M. Blagojevic. Efficient Cluster Mobility Support for TDMA-based MAC Protocols in Wireless Sensor Networks. *Submitted*.

- M. Nabi, M. Geilen, and T. Basten. On-Demand Data Forwarding for Automatic Adaptation of Data Propagation in WBANs. In *9th IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON 2012, Proceedings*, pages 250–258, Seoul, Korea. IEEE, 2012.

- M. Nabi, M. Geilen, and T. Basten. Demonstrating On-demand Listening and Data Forwarding in Wireless Body Area Networks. In *9th IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON 2012*, Invited demo, pages 82–84, Seoul, Korea. IEEE, 2012.

- M. Nabi, M. Blagojevic, M. Geilen, and T. Basten. Dynamic Data Prioritization for Quality-of-Service Differentiation in Heterogeneous Wireless Sensor Networks. In *8th IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON 2011, Proceedings*, pages 296–304, Salt Lake City, Utah, USA. IEEE, 2011.

- M. Nabi, M. Geilen, and T. Basten. MoBAN: A Configurable Mobility Model for Wireless Body Area Networks. In *4th International Conference on Simulation Tools and Techniques, SIMUTools 2011, Proceedings*, Barcelona, pages 168–177, Barcelona, Spain. ICST, 2011.

- M. Nabi, M. Geilen, T. Basten, M. Blagojevic, and T. Hendriks. A Robust Protocol Stack for Multi-hop Wireless Body Area Networks with Transmit Power Adaptation. In *5th International Conference on Body Area Networks, BodyNets 2010, Proceedings*, pages 77–83 , Corfu, Greece. ICST, 2010.

- M. Nabi, M. Blagojevic, M. Geilen, T. Basten, and T. Hendriks. MCMAC: An Optimized Medium Access Control Protocol for Mobile Clusters in Wireless Sensor Networks. In *7th IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON 2010, Proceedings*, pages 28–36, Boston, MA, USA. IEEE, 2010.

Publications not covered in the thesis

- M. Nabi, M. Geilen, and T. Basten. An Empirical Study of Link Quality Estimation Techniques for Disconnection Detection in WBANs. *Submitted*.

- M. Blagojevic, M. Geilen, T. Basten, M. Nabi, T. Hendriks Fast Performance Simulation for Gossip-based Wireless Sensor Networks. *Submitted*.

- R. Tavakoli, M. Nabi. TIGeR: a Traffic-Aware Intersection-Based Geographical Routing Protocol for Urban VANETs. In *77th IEEE Vehicular Technology Conference, VTC2013-Spring, Proceedings*, Dresden, Germany, 2013.

- M. Blagojevic, M. Nabi, M. Geilen, T. Basten, T. Hendriks, and M. Steine. Probabilistic Acknowledgment Mechanism for Wireless Sensor Networks. In *6th IEEE International Conference on Networking, Architecture, and Storage, NAS 2011, Proceedings*, pages 63–72, Dalian, China. IEEE, 2011.

- M. Nabi, M. Blagojevic, M. Geilen, T. Basten, and T. Hendriks. Configuring Multi-Objective Evolutionary Algorithms for Design-Space Exploration of Wireless Sensor Networks. In *4th ACM International Workshop on Performance Monitoring, Measurement and Evaluation of Heterogeneous Wireless and Wired Networks, PM2HW2N 2009, Proceedings*, pages 111-119. Tenerife, Canary Islands, Spain. ACM, 2009.

- M. Nabi, M. Blagojevic, M. Geilen, T. Basten, and T. Hendriks. Exploring a WSN Design Space using Genetic Algorithms. In *5th International Summer School on Advanced Computer Architecture and Compilation for Embedded Systems, ACACES 2009*, Poster abstracts, pages 327-328. Terrassa, Spain. HiPEAC Network of Excellence, Ghent, Belgium, 2009.

- M. Blagojevic, M. Nabi, M. Geilen, T. Basten, and T. Hendriks. Fast Simulation Methods to Predict Wireless Sensor Network Performance. In *6th ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks, PE-WASUN 2009, Proceedings*, pages 41-48, Tenerife, Canary Islands, Spain. ACM, 2009.