

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

No attribué par la bibliothèque

--	--	--	--	--	--	--	--	--	--

T H È S E

pour obtenir le grade de

DOCTEUR DE L'INPG

Spécialité : Informatique : Systèmes et Logiciels

préparée au laboratoire LSR – IMAG

dans le cadre de l'École Doctorale

Mathématiques, Sciences et Technologies de l'Information

présentée et soutenue publiquement par

Abdelmalik BACHIR

Le 29 Janvier 2007

**Optimizing Routing and Channel Access Protocols to Extend
the Lifetime of Wireless Sensor Networks**

Directeur de thèse : M. Andrzej DUDA

Co-encadrant : M. Dominique BARTHEL

Co-encadrant : M. Martin HEUSSE

JURY :

M. Jacques MOSSIÈRE,	Président
M. Christian BONNET,	Rapporteur
M. David SIMPLOT-RYL,	Rapporteur
M. Andrzej DUDA,	Directeur de thèse
M. Dominique BARTHEL,	Co-encadrant
M. Martin HEUSSE,	Co-encadrant

Dedication

To my parents

Acknowledgments

My first and deepest thanks go to my co-advisors: Dominique Barthel, Andrzej Duda, and Martin Heusse, for their able guidance and productive comments without which this thesis would not have been completed.

Dominique offered me an invaluable opportunity for doing research by accepting me within his research team at France Telecom R&D Labs. Despite his perpetually overbooked schedule, Dominique has always been available: not only he tolerated my time-consuming random musings, but also he spent countless occasions teaching me many important points, both in wireless communication and embedded programming. Dominique also taught me to look into the details and the feasibility of ideas.

In the LSR labs, Andrzej pushed me toward excellence by promoting high-quality research. He continuously challenged the feasibility of my ideas by always asking for real world experimentation to validate the mathematical analysis and simulation. Co-authoring papers with Andrzej has substantially improved my writing and presentation skills: I have learned how to make trade-offs between conciseness and comprehensiveness. I also want to thank Andrzej for his great input to this manuscript.

Martin has dedicated substantial and regular time for discussion and in-depth analysis. I always found Martin around when an issue occurred: his encouragements and support made me enjoy my stay in LSR labs. I just will not forget the repeated friendly discussions I had with him around cups of tea.

I am very grateful to Professors Jacques Mossière, Christian Bonnet and David Simplot-Ryl for having accepted to be in the jury of my thesis. Their feedback and comments provided me worthy ideas for further improvements and future work directions.

I have spent a wonderful time while being in France Telecom R&D labs. Since my arrival, very nice people facilitated my integration within them. During my stay in France Telecom I reported to many persons, Roland Airiau, Jean-Luc Leclercq, Vincent Gimeno and Marylin Arndt who all made all things possible for the accomplishment of my work. I also thank Patrice Senn and Yamina Allaoui, successive heads of the DIH/OCF then TECH/IDEA labs in which I did my thesis.

Many people offered me considerable help that accelerated the achievement of my work. Special thanks go to:

Laurent Granger and Sylvain Plancoulaine for their software developments and testbed installations;

Fredéric Evennou for his piece of Matlab code that makes coffee "‘qui fait le café’";

Mischa Dohler and Benoit Miscopein for manuscript proof-reading and insightful feedback on MAC and PHY layers;

Christian Lereau for his in-depth course on budget link;

Gabriel Chegaray for the discussion I had with him on 802.15.4 and for his friendly encouragements;

Throughout the course of my thesis, I spent fun times with many friends. Thanks go to: Rudy, Ludovic, Lahouari, Mohammed, David, Philippe, Jean, Thomas, Apostolos, Chawki, Mathieu, Jerome, François, ... I am sure I have missed a lot of names, so please be indulgent.

Finally, my innermost thanks go to my parents and my family for their endless support and special help.

Contents

1	Introduction	1
I	Context and State of the art	9
2	Channel Access Protocols	11
2.1	Main Sources of Energy Dissipation	11
2.2	Energy Efficient Techniques	12
2.2.1	Common Active/Sleep Schedules	12
2.2.2	Preamble Sampling	18
2.3	Standardization	23
2.4	Conclusions	24
3	Routing Protocols	27
3.1	Characteristics of Routing in WSN	27
3.1.1	Data-Centric Communication	27
3.1.2	Scarce Energy and Computation Resources	27
3.1.3	Other Features	28
3.2	Data-Centric Routing	28
3.2.1	Directed Diffusion	28
3.2.2	Variants of Directed Diffusion	30
3.2.3	Discussion	33
3.3	Energy-Efficient Routing	34
3.3.1	Reducing Protocols Overhead	34
3.3.2	Using Explicit Energy-Related Metrics	35
3.4	Conclusion	39
II	Contributions	41
4	Coping with Collisions	43
4.1	Inefficiency of Existing Solutions for Sensor Networks	43
4.2	Modeling Hidden and Visible Nodes Collision	44
4.2.1	System Model	44
4.2.2	Calculating the Number of Hidden and Visible Nodes	47
4.2.3	Numerical Results for Different Radio Technologies: Bluetooth, Zig-Bee, WaveLAN	47
4.3	Reducing Collisions	48
4.3.1	Tuning Carrier Sense Threshold	49

4.3.2	Adjusting Contention Window	50
4.4	Evaluation	52
4.5	Conclusion	53
4.6	Appendix	54
4.6.1	Calculation of the Surface of Hidden and Visible Nodes	55
5	Reducing Overhearing by Means of Abstract Frames	57
5.1	Abstract Frames	57
5.1.1	Basic Idea	57
5.1.2	Discussion	59
5.2	Theoretical Performance	59
5.2.1	Abstract Frames with an Ideal MAC	59
5.2.2	Abstract Frames with SMAC	60
5.3	Simulation	62
5.3.1	Simple Star Topology	63
5.3.2	Random Topology	65
5.4	Conclusion	66
6	Micro-Frame Preamble MAC	67
6.1	Protocol Description	67
6.1.1	Avoiding the Reception of the Whole Preamble	68
6.1.2	Avoiding the Reception of Irrelevant Data	68
6.1.3	Other benefits	69
6.2	Theoretical Evaluation	69
6.2.1	Reception Cost	70
6.2.2	Channel Sampling Cost	72
6.2.3	Transmission Cost	72
6.2.4	Modeling Node Lifetime	73
6.2.5	Comparisons	75
6.3	Simulations	75
6.3.1	Network-Scope Evaluation	78
6.3.2	Node-Scope Evaluation	81
6.4	Implementation	81
6.4.1	Feasibility of MFP on Existing Radios	81
6.4.2	Microcontroller Overhead	86
6.4.3	Possible Improvements	86
6.5	MFP Compared with Other Protocols	87
6.6	Conclusions	88
7	Evaluating Link Cost and Reliability of Frame Preamble Protocols	89
7.1	Frame Preamble MAC Protocols	89
7.1.1	Transmission Schemes: MFP vs. DFP	89
7.1.2	Reception Schemes: Persistent vs. Non Persistent	90
7.2	System Model	92
7.2.1	Problem Statement and Assumptions	92
7.2.2	Reliability	92

7.2.3	Transmission Cost	92
7.2.4	Reception Cost	93
7.3	Evaluation	93
7.3.1	Non-Persistent DFP	93
7.3.2	Non-persistent MFP	94
7.3.3	Persistent DFP	94
7.3.4	Persistent MFP	96
7.3.5	Numerical Examples	97
7.4	Conclusion	99
8	$O(1)$-Reception Routing	101
8.1	Overview of $O(1)$ -Reception Routing	102
8.2	Problem Statement and System Model	103
8.3	Approximate Solution: Heuristic Functions	105
8.4	Exact Solution: Synthetic Function	106
8.5	Analytical Evaluation	109
8.5.1	Worst Case Interest Propagation Delay	109
8.5.2	Vulnerable Routes Rate	109
8.6	Simulations	111
8.7	Conclusion	116
9	Conclusions	119

List of Figures

1.1	Global Vision of Sensor Networks	1
2.1	SMAC alternates turning on and off the radio. SMAC splits the active period into two sub-periods: one for exchanging sync messages and the other for exchanging data messages. Data message exchange may require RTS, CTS and ACK utilizations.	13
2.2	TMAC downsizes active period lengths to further save energy. The arrows indicate transmitted and received frames. Both TMAC and SMAC move the traffic that comes during a sleep period to the subsequent active period. TMAC prematurely ends an active period if no traffic occurs for a duration of TA	15
2.3	The transmission of FRTS aims at keeping node D awake. FTRS control frames make it possible for TMAC to achieve a transmission over three hops within a single active period. This technique reduces the end-to-end latency. .	16
2.4	Preamble sampling technique	18
2.5	Lifetime of nodes according to different check intervals and traffic loads. . . .	20
2.6	WiseMAC improves on preamble sampling energy savings through the use of short preambles. In WiseMAC, each transmitter knowing the wakeup time of the receiver sends its transmission just on time to meet the receiver wakeup. The transmitter may use a short preamble to cover clock drifts about the receiver's wakeup time.	21
3.1	Interest propagation.	31
3.2	Gradient setup.	31
3.3	Gradient reinforcement.	32
3.4	Data path.	32
3.5	An example showing how diffusion tolerates duplicate node identifiers.	33
3.6	Operation of rumor routing.	36
3.7	An example with nodes having different remaining energies and different link costs.	37
4.1	Transmission, reception, and interference ranges.	45
4.2	Hidden nodes area with a Free Space model.	48
4.3	Hidden nodes area with a 2-Ray Ground Reflection model.	49
4.4	Contention Window in function of collision probability for the ZigBee radio with antenna height of 0.1m.	51
4.5	Collision probability due to hidden nodes.	52
4.6	Collision probability due to contention.	53
4.7	A reception system	54

4.8	Transmission, reception, and interference ranges.	55
5.1	Avoiding redundant frame reception by means of Abstract Frames.	58
5.2	Operation of SMAC with and without Abstract Frames.	61
5.3	Lifetime extension according to various data frame sizes.	64
5.4	Lifetime extension according to various traffic loads	65
5.5	Lifetime extension according to various number of neighbors	65
5.6	Lifetime extension in function of the number of nodes in the network.	65
6.1	Microframe structure as implemented in the CC2500 module. Field sizes, in bytes, are in parentheses.	68
6.2	Avoiding reception of irrelevant preamble.	68
6.3	Avoiding reception of an irrelevant data frame.	69
6.4	Transmission of microframes in the preamble with a gap between two consecutive microframes.	70
6.5	Normalized lifetimes of a node using MFP in function of different inter microframes gaps and various traffic rates.	74
6.6	Normalized lifetime of nodes for different check intervals. Microframes are transmitted without inter microframe gaps, i.e. $T_S = 0$	76
6.7	Chosen random topology: Node 0 is the sink, Nodes 1 through 5 are the sources, the other nodes (6 through 29) are intermediate nodes that relay traffic.	77
6.8	The mean lifetime of nodes for various check intervals. It is measured in terms of the number of relevant frames that passed through the node per 1 joule. The lifetime is averaged over all the nodes.	78
6.9	The mean lifetime extension of MFP compared to LPL.	79
6.10	The mean collision rates.	80
6.11	Message collision in ns-2 simulation. Messages 1 and 2 are transmitted by nodes that are out of transmission range of each other — the collision occurring at the receiver is caused by the hidden node problem.	80
6.12	Current consumption during periodic channel sampling done by the radio module. We have performed these current measurements only on the radio module.	82
6.13	Current consumption during frame forwarding using MFP taken at the radio and the microcontroller together.	83
6.14	Details of a data frame reception. Measurements are performed only on the radio.	83
6.15	Details of a transmission (microframes and a data frame). Measurements are performed only on the radio.	84
6.16	Distribution of the energy consumption of both the radio and the microcontroller for the different MFP operations.	85
6.17	The percentage of lifetime extension in function of different values for the current drained by the microcontroller in Idle and Active modes. Reducing the current drained in the Active mode only offers a negligible gain, whereas reducing that of the Idle mode significantly increases the lifetime.	87

7.1	MFP	90
7.2	DFP	91
7.3	Average transmission duration	97
7.4	Average reception duration	97
7.5	Average total duration	98
7.6	Link reliability	98
8.1	Principles of energy-delay mapping technique.	103
8.2	Heuristic Mapping Functions.	106
8.3	Energy Levels.	107
8.4	The worst case illustrated with two routes.	108
8.5	The rate of vulnerable routes in function of the probability that a node is vulnerable. $ \mathcal{R} $ is the number of disjoint routes between the source and the destination.	111
8.6	Random Network.	112
8.7	Star Network.	112
8.8	Lifetime extension according to γ with the Ideal-filter MAC protocol. The value of m is set to 4.	113
8.9	Percentage of improvement with $\gamma = 1.0$ for a star network.	114
8.10	Percentage of improvement with $\gamma = 1.0$ for a random network.	114
8.11	Average interest propagation delay with $\gamma = 1.0$ for a star network.	115
8.12	Average interest propagation delay with $\gamma = 1.0$ for a random network.	115
8.13	Lifetime extension for each node in the random topology.	116

List of Tables

3.1	Summary of the three variants of Directed Diffusion. Numbers indicate the sequence of operations	30
4.1	Notation for the analysis	46
4.2	Channel models used for the analysis	46
4.3	Radio Parameters	48
6.1	Performance for chosen nodes.	81
6.2	Measured current consumption of CC2500	82
8.1	Notation	109

1 Introduction

Sensor wireless networks are composed of a potentially large number of sensor nodes communicating via wireless radio interfaces. Unlike nodes used in traditional wireless networks such as cell phones, PDAs, and laptops, sensor nodes are particularly small-sized, low power, and low cost. They include a minimal number of essential elements for operating over long periods of time: sensing the physical world for some meaningful data, processing and communicating the information to end users. Sensor networks are usually application-driven and perform specific tasks so they can support a specific class of chosen applications instead of providing a general-purpose application interface such as the Internet. Sensor network applications are centered on sensing, which is the ability of detecting environmental conditions, such as temperature, humidity, and chemicals, in regions surrounding nodes. The role of sensor nodes is to monitor these phenomena and send reports describing them to a special node, usually referred to as sink node, which gathers data and issues commands accordingly. The sink node may also serve as a gateway for connecting sensor networks to other networks such as the Internet (as shown in Figure 1.1). Interconnecting sensor networks with other networks leads to large-scale information dissemination to other entities, such as Internet or cell phone users.

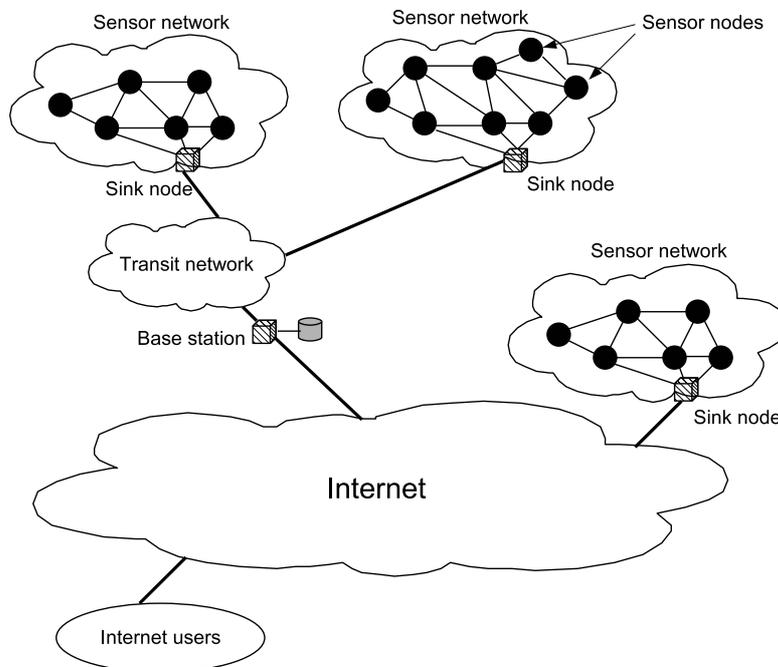


Figure 1.1: Global Vision of Sensor Networks

1 Introduction

Sensor networks promise a wide variety of applications and services: they can be used for example to perform surveillance [1], meter reading [2], habitat monitoring [3, 4], disaster management [5], home and building automation [6], etc. With sensor networks, such applications will generate substantial benefits, both economical and ecological. As an example, the use of sensor networks for controlling the temperature of buildings may save billions of US dollars and cut carbon gas emission by the order of million tons [6].

The wide range of applications for sensor networks makes the deployment of sensor nodes fairly diversified. Nodes may be manually installed by humans or robots, embedded in construction materials, or spread in nature by aircrafts. These various forms of deployment require that sensor nodes be easy to handle and have a low cost. Sensor nodes are designed to meet such requirements: they use batteries for power supply, communicate via wireless radio interfaces, and manage to communicate without a fixed communication infrastructure. Sensor networks compensate for the lack of communication infrastructure by organizing themselves in a multihop topology. In such an organization, nodes cooperate by relaying messages to guarantee communication between those that cannot communicate directly.

In some cases, a pre-defined subset of nodes may form a data transport infrastructure. In those cases, infrastructure sensors benefit from more or infinite energy. Operating such networks is straightforward: sensors send their packets to the nearest infrastructure nodes that relay them toward the sink. In this work, we consider the case where all nodes have equal and limited resources, and thus must share the burden. However, we claim that many of the proposals presented hereafter are relevant even if an infrastructure is present.

The success of sensor networks depends on their ability to meet three essential requirements: sufficient quality of service, easy maintainability, and long lifetime. Sensor networks must guarantee and maintain a required quality of service they provide to applications and users for long periods of time. However, the quality of service in sensor networks may differ from usual parameters in traditional networks such as bandwidth and packet delivery ratio. Rather, it considers other parameters that are related to the information per se such as the quantity of information extracted from the gathered data, the confidence in the extracted information, etc. In some specific applications such as surveillance, the end-to-end communication delay is also among the considered parameters.

As both sensor networks and environments in which they are deployed may experience some changes that alter the correct operation of applications, e.g. depleted batteries, failed nodes, jammed communication channels, etc., sensor networks should provide maintenance mechanisms so that they continuously meet quality of service requirements. Maintenance mechanisms may be internal and external. The goal of maintenance is to extend the duration of the correct operation of the network according to the required quality of service. This duration is also referred to as the network lifetime.

The scarcity of processing, memory, and energy resources of sensor nodes makes meeting the cited above requirements fairly challenging. The processing unit typically used consists

of a tiny microcontroller with a reduced memory size and running at low speeds. For example, a widely used microcontroller is MSP430F16x with a 16-bit instruction set that runs at speeds ranging from 4 to 8 MHz and uses a 10KB RAM. Such limited microcontrollers only tolerate low complexity and low memory usage programs. The challenge of operating systems designed for sensor networks, such as TinyOs [7], is to minimize complexity while providing useful interfaces and a modular architecture.

Minimizing the complexity and the size of programs is only one of the various design challenges resulting from the scarce processing capabilities. The limited energy supplies generate another important challenge, which is minimizing energy consumption. Solutions for reducing energy consumption rely on switching off the maximum of active components (microcontroller, radio chip, sensing devices) whenever possible. A simple comparison between the energy consumption of the active modes of these components show that the radio chip is the most energy consuming. For example, an active low power radio chip (CC2500) may consume up to 10 times more energy than an active low power microcontroller (MSP430). Therefore, most of the proposed solutions for this problem aim at maximizing the time during which the radio is off [8, 9, 10, 11, 12].

As the radio chip acts as a communication interface, switching it off makes the node unable to communicate and receive frames, which may alter the correct operation of the whole network because some nodes with their radios switched off may miss frames destined to them. This problem, referred to as deafness, requires a suitable network organization. Coping with deafness in infrastructure-based wireless networks is not as complex as in infrastructure-less networks. Many protocols like the IEEE 802.11 PSM (Power Save Mode) and the IEEE 802.15.4 rely on the existence of powerful nodes composing the infrastructure and do not have constraints on energy. These nodes, referred to as access points in the IEEE 802.11 PSM and as coordinators in the IEEE 802.15.4, do not need to switch their radios off. Other nodes that are energy-constrained spend large periods of time with the radio switched off. They wake up only for a short time to communicate with the infrastructure nodes (access points or coordinators).

The organization for avoiding deafness in such networks considers only the energy constrained nodes because sending messages from energy constrained nodes to coordinators do not cause problems, Coordinators being permanently active. However, sending messages from coordinators to energy-constrained nodes may be subject to deafness. Two solutions can be used for solving these problems: polling or synchronization. In polling, each energy constrained node polls the Coordinators with whom it is associated asking for potential messages. If the Coordinator has messages for the node, it sends them directly to the node right after the poll message. In the method based on synchronization, energy-constrained nodes wake up periodically according to a predefined schedule. During the association phase, the Coordinator gathers the wake up schedules of these nodes. Thus, when a Coordinator wants to send a message to an energy constrained node, it waits until the node wakes up to send the message. Note that the infrastructure also solves the deafness problem between two energy constrained nodes, because the infrastructure is involved in relaying messages from one energy constrained node to another.

1 Introduction

The lack of a fixed infrastructure in multihop sensor networks amplifies the problem of deafness. In sensor networks, nodes cannot afford to keep their radios on in permanence to avoid deafness, because this would result in high energy waste as the communication between nodes is not frequent—most of sensor network applications usually generate low traffic. In these situations, the communication channel is expected to be idle most of the time and idle listening, the state of a node that listens to an idle channel, is the major source of energy waste in sensor networks. Therefore, most of the research work on energy saving in sensor networks has focused on reducing idle listening while avoiding deafness. From a layered architecture view, the most suitable place for solving the problem of deafness and idle listening is the channel access layer that organizes the contention to the common radio channel of neighbor nodes.

There are two major classes of channel access protocols for sensor networks that reduce idle listening: those using common active/sleep periods and those using preamble sampling techniques. In the former, nodes define two types of periods: active and sleep. During the active periods nodes communicate and they switch their radios off during the sleep periods to save energy. To avoid deafness, the active and sleep periods are common to at least neighbor nodes. In preamble sampling protocols, nodes keep their radios off most of the time and periodically wake up for a short time to check whether there is an ongoing transmission on the channel. If a node detects a transmission, it keeps its radio on to receive the data frame. To avoid deafness, nodes precede each data frame with a preamble long enough to make sure that all the nodes will wake up at least once during the preamble. Thereby, potential receivers are ensured not to miss the data frame.

The performance of these channel access protocols are promising, because they reduce idle listening, however, they still may be improved. Protocols based on common active/sleep periods require that nodes exchange messages to synchronize common periods, which increases the overhead and thus reduces their energy savings. Protocols based on preamble sampling do not experience the same drawbacks, but they suffer from another form of overhead—the use of a long preamble. As a long preamble precedes each data frame, the energy drained per transmitted and received frame becomes very high. In addition, collisions and transmission errors become very costly because significant energy drained in transmission and in reception of frames that get corrupted is just wasted.

By organizing the interaction between a node and its neighborhood, channel access protocols only reduce energy waste at a local level. Although these protocols achieve high energy savings, they are not sufficient alone to meet stringent lifetime requirements. The definition of the lifetime in a sensor network also depends on applications: it may be the time until the first node runs out of energy, the time until the network is partitioned, or the time until all the nodes run out of energy. Such different situations emphasize the need for a global network organization to extend the lifetime of a sensor network. From a layered architecture view, the global network organization involves routing protocols.

To extend the lifetime of sensor networks, routing protocols consider three improvements:

two of them are generic whereas the third one depends on the definition of the lifetime. The two first approaches decrease the overall energy consumption by reducing the number of exchanged messages [13, 14, 15, 16, 17] and selecting routes that consume a minimum energy per transmitted packet [18]. The third approach builds upon cooperation between nodes according to the considered lifetime definition. For example, if the lifetime is the time until the network is partitioned, nodes should identify the critical nodes sustaining the connectivity of the network to avoid overusing them and thus to delay their energy exhaustion [19, 20, 21, 22, 23].

The three improvement approaches applied to routing protocols are generally antagonistic: cooperation between nodes increases the number of the exchanged messages and routes that avoid overusing critical nodes do not necessarily consume the minimum of energy. The challenge is thus to find a way to benefit from all of them within a single routing protocol.

Optimized routing and channel access protocols extend the lifetime of sensor networks. However, considering each layer independently of the other may result in sub-optimal solutions. For example, some energy-efficient channel access protocols such as SCP [24], WiSeMAC [25], X-MAC [26], CSMA-MPS [27], STEM-B [28], and WOR [11] have different energy costs for unicast and broadcast transmissions: unicasts cost less than broadcasts. A good routing protocol should be aware of this characteristics so that it gives more priority to reducing broadcast communications. Therefore, to obtain high performance, the design of protocols should follow a cross-layer approach that reinforces the interactions between network protocol layers [29]. At the same time, cross-layer designed protocols should still have a modular architecture to ensure their maintainability and interoperability, which are among the essential elements for the success of solutions [30].

In this dissertation, we improve both routing and channel access protocols while taking into account their interactions. Our protocols fit into a layered architecture and thus they are interoperable with other protocols. Their joint use yields better performance. At the channel access layer, we reduce energy waste through dealing with collisions and overhearing. At the routing layer, we extend the lifetime of sensor networks by reducing the overhead of protocols while selecting optimal routes. The reduction of the overhead in our routing protocol is effective only if the channel-access protocol beneath enables filtering of irrelevant messages, a feature enabled by our channel-access protocols.

Summary of contributions and thesis organization

This dissertation is structured in nine chapters divided into two parts: the state of the art and contributions. The state of the art part presents background on extending the lifetime of sensor networks at the channel access and routing layers. The contribution part contains our proposals for optimizing the performance of wireless sensor networks.

In Chapter 2, we dwell on the most significant low-power channel access protocols. We show that the main challenge of these protocols is to maximize the time during which the radio is off while ensuring the correct operation of the network. We present the operation

1 Introduction

of two prominent classes of protocols in detail: those using common active/sleep periods and those using preamble sampling. We discuss the advantages and drawbacks of both of these classes of protocols. We show that although preamble sampling protocols yield better energy savings under low traffic-load conditions, they still present limitations resulting from the use of the long preamble. We present methods for alleviating these limitations in the contribution part.

In Chapter 3, we present the most important approaches used for extending the lifetime of sensor networks at the routing layer. We show that routing protocols aim at achieving two objectives: minimizing the overall energy drained in routing packets, and avoiding premature energy exhaustion of critical nodes. Minimizing the overall energy consumption requires both reducing the overhead of protocols, mainly the number of exchanged messages, and selecting minimum energy consumption routes. Avoiding premature energy exhaustion of critical nodes requires cooperation between nodes to identify the critical ones. We show that these approaches are antagonistic. First, minimum energy consumption routes do not necessarily avoid the premature energy exhaustion of critical nodes. Second, both selecting minimum energy consumption routes and avoiding premature energy exhaustion of critical nodes require exchanging routing messages, which increases the overhead. We show that there is a class of protocols solving the first antagonistic elements by using a combined metric to make a trade-off between selecting minimum energy consumption route and avoiding premature energy exhaustion of critical nodes. We show that protocols of that class still present shortcomings mainly resulting from their overhead.

Chapter 4 and onwards start our contributions. In Chapter 4, we revisit the problem of collisions in multihop networks. We point out that solutions currently in use to reduce collisions, such as RTS/CTS, are unsuitable for sensor networks. We distinguish collisions caused by visible nodes from those caused by hidden nodes. We model both types of collisions and derive a closed-form formula to determine the probability of these collisions according to various channel models and various radio chip specifications. We provide two solutions for lowering these collisions. The first one based on tuning the carrier sense threshold saves a substantial number of collisions through reducing the number of hidden nodes. The second one based on adjusting the contention window reduces the probability that two transmissions overlap and thus reduces both hidden and visible nodes collisions. We evaluate the performance of proposed solutions through simulations.

In Chapter 5, we deal with overhearing, which is another source of energy waste. Overhearing occurs when a node drains energy when receiving irrelevant frames, e.g. unicast frames addressed to other nodes. We identify another type of irrelevant frames: redundant broadcast frames. In a flooding, a node may receive multiple copies of the same data frame from several neighbors. To eliminate such receptions, we propose to use abstract information — a digest of the data frame. The abstract information is sent in a small frame before the data frame. When a node receives the abstract information, it knows whether the subsequent data frame is redundant or not. If the data frame is redundant, the node switches its radio off to save energy. The abstract information can be used with a large set of MAC protocols. In Chapter 5, we show how it can be used with protocols using common

active/sleep periods and in Chapter 6, we show its use with preamble sampling protocols. In both chapters, we evaluate analytically and via simulation the performance gains obtained with the use of abstract information for filtering redundant messages.

In Chapter 6, we identify another type of overhearing that is specific to preamble sampling protocols: a node that wakes up and finds a preamble being transmitted on the channel keeps receiving this preamble until it receives the data frame. This reception is not necessary as the preamble does not carry any useful information. To avoid it, we replace the preamble by a series of small frames called micro-frames. Each micro-frame carries information about the data frame: its destination address, the time of its transmission, and an abstract of its contents. When a node wakes up during an ongoing transmission, it receives a micro-frame from which it learns whether the subsequent data is relevant and when it will be transmitted. If the data frame is relevant, the node switches its radio off to avoid receiving the remainder of the series of micro-frames and switches it back on only to receive the data frame. We show, through analysis and simulation, that the use of micro-frames results in a substantial energy saving. We also show that it is feasible with radio chips currently in use and it does not require additional circuitry.

In Chapter 7, we generalize the idea of micro-frames used instead of the preamble and propose the frame-preamble channel-access protocol. According to the policy used by transmitters and receivers, the frame preamble protocol generates many variants. For the transmission, we consider two policies: in the first, the preamble is replaced by micro-frames and in the second one it is replaced by data-frames. For the reception, we also consider two policies: in the first one, a node does not persist in reception when it cannot receive a correct preamble frame (data or micro-frame) within a predefined time and in the second one, a node persists until it receives a frame or the channel becomes idle. These various policies generate four variants of preamble frame protocols, namely pMFP (persistent Micro Frame Preamble), npMFP (non persistent MFP), pDFP (persistent Data Frame Preamble), and npDFP (non persistent DFP). We model the operation of these variants assuming a channel subject to transmission errors and derive closed form formulas for the reliability and the energy cost of each variant. We show that higher reliability is required as it reduces the energy cost through reducing the number of retransmissions that are very costly.

In Chapter 8, we propose a routing protocol that contributes to extending the lifetime of sensor networks. Our proposal reduces the overhead of the routing protocol while using a combined metric that makes a trade-off between selecting minimum energy consumption routes and avoiding premature energy exhaustion of critical nodes. Our protocol, called $O(1)$ -reception routing, eliminates the need of receiving multiple routing messages traditionally required for selecting the best route. With $O(1)$ -reception routing, a node is able to select the best routes based on only one routing message reception. The key idea of our protocol is the energy-delay mapping technique that maps energy information into propagation delay so that the first received routing message indicates the best route. To be effective, the $O(1)$ -reception routing should be used on top of a channel access protocol that enables filtering irrelevant messages so that redundant routing messages are actually not received. The filtering of redundant messages is possible thanks to the idea of the abstract

1 Introduction

information presented in Chapters 5 and 6. We show through simulation that the joint use of $O(1)$ -reception routing with a micro-frame preamble channel-access protocol achieves significant lifetime extensions compared to the work presented in the state of the art.

In Chapter 9, we conclude this dissertation and present our view on future research directions.

Part I

Context and State of the art

2 Channel Access Protocols

The role of channel access protocols, also referred to as the MAC (Medium Access Control) protocols, is to organize the access to the common wireless channel. They set up rules that determine when a node can transmit and when a node should listen to the channel to receive frames. Traditional MAC protocols [31] aim at improving network throughput, ensuring fairness among nodes, and minimizing end-to-end communication delay. These metrics are also considered in sensor networks, but with less importance compared to saving energy — many protocols [32, 33] have been specially designed to minimize energy consumption. In this chapter, we first discuss the main sources of energy dissipation at the MAC layer. Then, we describe two¹ major classes of protocols: those using common active/sleep periods and those using preamble sampling. We discuss their drawbacks and advantages and proposed improvements to them. We also touch upon the IEEE 802.15.4 standard [38] and discuss its suitability for multihop sensor networks.

2.1 Main Sources of Energy Dissipation

One of the main objectives of MAC protocols for wireless sensor networks is to minimize power consumption while providing reliable low-rate data transmission. Previous studies have identified multiples sources of energy dissipation at the MAC layer, the most important being the following [39]:

Idle Listening: it happens when a node does not know when it will be the receiver of a frame so that it keeps its radio on while listening to the channel waiting for potential data frames. The amount of energy wasted whilst the radio is on is considerable even when it is neither receiving nor transmitting frames.

Collisions: they may happen when a node is within the transmission range of two or more nodes that are simultaneously transmitting so that it does not capture any frame. The energy drained in the transmission and reception of collided frames is just wasted. We devote Chapter 4 to this problem and present some solutions for reducing collisions.

Overhearing: it happens when a node drains energy receiving irrelevant frames or signals. Irrelevant frames may be for example unicast frames destined to other nodes. Overhearing such irrelevant unicast frames can be avoided through a filtering based on their destination addresses. In Chapters 5 and 6, we identify and propose solutions

¹We are fully aware of the amount of work conducted in the area of MAC protocols for wireless sensor networks [32, 31, 33, 34, 35, 36, 37]—an exhaustive referencing is hence beyond the scope of this chapter. Our goal in this chapter is to present the most representative protocols closely related to our work discussed in the next chapters.

to two other forms of overhearing: the reception of redundant broadcast messages and the reception of the long preamble in preamble-sampling protocols.

Protocol Overhead: protocol overhead may result in energy waste, for example, the energy drained in transmitting control frames—RTS and CTS control frames used in some protocols do not carry any data although their transmission consumes energy. Note that RTC/CTS exchange incurs high overhead, 40% to 75% of the channel capacity, because data frames are typically very small in sensor networks [40, 41].

2.2 Energy Efficient Techniques

Sensor network applications usually generate low traffic load, thus the communication channel is expected to be idle most of the time. Under such circumstances, idle listening is the most significant source of energy dissipation. Without any specific energy management, nodes waste considerable amounts of energy as they keep their radios on for large time intervals while listening to an idle channel. To mitigate idle listening, energy-efficient MAC protocols make nodes sleep for long periods of time instead of being active permanently [32, 33]. These MAC protocols define a *duty-cycle parameter* to control the ratio of the activity period to the sleep period.

Duty cycles of 1% or less substantially reduce energy consumption. However, if a node wakes up only at some chosen instants to mitigate idle listening effects, then there is a need for a method to detect transmissions of other nodes so as to avoid deafness. Deafness happens when a node transmits a message to another node that is sleeping. There are two ways for avoiding deafness. In the first approach used in protocols like SMAC (Sensor MAC) [39], TMAC (Timeout MAC) [42] and others, nodes synchronize on a common sleep/wakeup schedule by exchanging synchronization messages to set their sleep/wakeup schedule. The second approach used in protocols like WiseMAC (Wireless Sensor MAC) [25] and BMAC (Berkeley MAC) [41], does not set up a common schedule for sleep and wakeup periods to avoid synchronization overhead and to further reduce idle listening in lightly loaded networks.

In this section, we dwell on these two approaches and present the major related contributions.

2.2.1 Common Active/Sleep Schedules

Basic Idea

SMAC [39] is a seminal work in this area, we take it as a representative protocol. SMAC copes with idle listening by repeatedly putting nodes in active and sleep periods. Nodes turn off their radios in sleep periods to save energy and they turn them on in active periods to exchange messages. Active periods are of fixed size whereas the length of sleep periods depends on a predefined duty-cycle parameter.

SMAC deals with deafness by making nodes share common active periods. Using common active periods requires synchronization establishment and maintenance among nodes. SMAC splits active periods into two sub periods: one for exchanging *SYNC* messages and

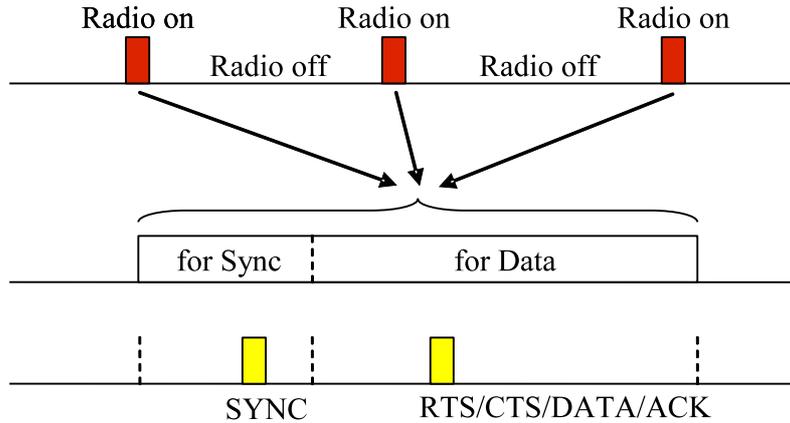


Figure 2.1: SMAC alternates turning on and off the radio. SMAC splits the active period into two sub-periods: one for exchanging sync messages and the other for exchanging data messages. Data message exchange may require RTS, CTS and ACK utilizations.

the other one for exchanging data messages (as shown in Figure 2.1). Each sub period is divided into mini-slots². In both these sub-periods, nodes contend for the channel in a similar way to the IEEE 802.11 DCF [43]: nodes perform a carrier sense first and then transmit in the next mini-slot if the channel is sensed free.

Each node using SMAC should have a schedule according to which it determines when it turns on its radio and when it turns it off. When deployed for the first time, a node starts by looking whether there are existing schedules in the network. The node keeps continuously listening to the channel for a duration of a least one active period plus one sleep period. If the node receives a *SYNC* message, then it adopts the schedule carried by that message. However, if it does not receive any *SYNC* message, the node chooses its own schedule and follows it. Once a node has a schedule, it disseminates it throughout the neighborhood by broadcasting a *SYNC* message with that schedule. Some of the node's neighbors receive the *SYNC* message; these neighbors adopt the schedule and continue disseminating it throughout the network.

Nodes that follow the same schedule form a virtual cluster. A network is most likely to contain several virtual clusters. Transmission errors, collisions, large end-to-end delays, simultaneous self schedule selection and other factors may result in different *SYNC* messages with different schedules being transmitted in the network. Some nodes may receive several *SYNC* messages with various schedules. These nodes are called *border nodes*. Border nodes should adopt all³ the schedules they receive and thus keep their radios on during all the active periods determined by these schedules. Border nodes sustain network connectivity by ensuring message passing from one cluster to another.

Applications may want to send messages while nodes are in sleep periods. SMAC post-

²SMAC implementation in ns2.27 uses 31 mini slots for the *SYNC* sub-period and 63 mini-slots for the data period.

³Actually, some implementations suggest that border nodes adopt only some schedules to reduce the time during which the radio is on. Although this further saves energy, it may cause network fragmentation as some virtual cluster may be isolated.

2 Channel Access Protocols

pones these transmissions until the next active period. As nodes sleep most of the time, strong contention for the channel is expected in the active periods. Specifically, nodes are implicitly synchronized at the beginning of active periods and there is a significant chance they simultaneously access to the channel at the beginning of active periods. SMAC copes with this kind of collisions by having nodes backoff for a random duration before transmission. SMAC also copes with collisions through the use of traditional mechanisms such as RTS/CTS exchange and virtual listening according to NAV (Network Allocation Vector). The NAV contains a value that tells the node if there is an ongoing transmission and, if so, when it ends. NAV sets this value from overhearing headers of RTS, CTS and Data frames — these headers carry information about transmission durations.

In SMAC, nodes do not transmit long messages in a single packet because this costs retransmission of the whole packet in case of a collision, even when only a few bits are corrupted. Instead, nodes fragment each long message into many independent small packets and transmit them in a burst. Nodes use RTS/CTS only before transmitting the first small packet. The RTS/CTS exchange, in this case, reserves the channel for the whole burst duration instead of reserving it only for the subsequent packet as usual. Although this is unfair from a per-hop MAC level, it saves the energy of using RTS/CTS exchange before each small packet transmission.

Discussion

The use of common/sleep periods of a fixed size generates the following problems:

Rigidity: determining the optimal size of active periods requires taking into account two parameters: idle listening and collisions. Short active periods reduce idle listening, but they increase contention and thus collision rates. Long active periods do the opposite, they reduce contention at the cost of increased idle listening. SMAC uses a fixed pre-calculated size for active periods that is optimized for an expected workload. This makes SMAC rigid as nodes have no means to dynamically change their duty-cycle to meet time-varying or spatially non-uniform traffic loads. Note that variable workloads are expected in sensor networks as some nodes may be involved in relaying traffic more than others. For instance, nodes that are closer to a sink are most likely to relay more traffic than the other nodes.

Sleep Delay: sleep periods do save energy; however, they introduce extra end-to-end delay called *sleep delay*. Sleep delay increases communication latency in multihop networks as intermediate nodes on a route do not necessarily share a common schedule.

Further Improvements

Mitigating Rigidity:

TMAC [42] follows up on the basic idea introduced by SMAC that consists in using common active/sleep schedules: nodes determine their active/sleep schedules in a way similar to SMAC. TMAC alleviates the SMAC's rigidity by proposing an adaptive duty-cycle in which the duration of active periods is no longer fixed but varies according to the traffic. The key idea of TMAC consists in making a node predict channel activity

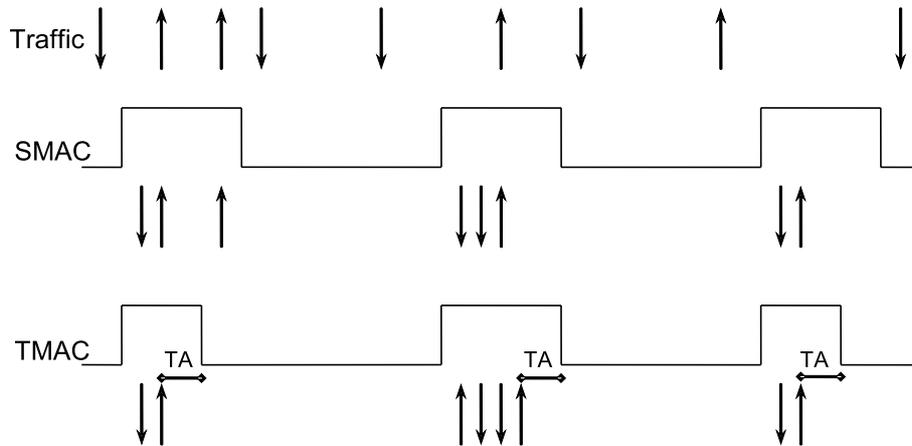


Figure 2.2: TMAC downsizes active period lengths to further save energy. The arrows indicate transmitted and received frames. Both TMAC and SMAC move the traffic that comes during a sleep period to the subsequent active period. TMAC prematurely ends an active period if no traffic occurs for a duration of TA .

during an active period so that it can switch its radio off before the active period ends, in case it does not expect any traffic. Figure 2.2 shows the overall operation of TMAC and its difference compared to SMAC.

By downsizing active period lengths, TMAC saves more energy than SMAC. The proportion of this saving depends on the amount of time cut back on the initial active period duration; the more nodes sleep during active periods, the larger the saving. To optimize the sleep period durations, TMAC moves all communications to a burst at the beginning of active periods. Therefore, a node can determine that there will be no communications in the remainder of an active period if no *activation event* occurs within the duration of TA . An activation event may be, for instance, the reception of a frame or sensing some noise considered as collision. The minimum duration of TA should be long enough to span the maximum contention duration and the RTS/CTS exchange (see Figure 2.3).

By having nodes ending their active periods prematurely, TMAC partially breaks the synchronization among nodes within a virtual cluster, which leads to the *early sleep problem*. The early sleep problem happens when a third hop node, supposed to be the next relay of an ongoing transmission, prematurely goes to sleep. TMAC copes with this by using the FRTS (Future Request To Send) frames sent to the third hop node before its TA timer expires. Thus, the third hop node stays active and then receives the next transmission right away instead of receiving it in the next active period in case FTRS was not used.

In variable workloads, TMAC saves about five times more energy than SMAC does. However, this is achieved at the cost of an increased latency and thus reduced throughput. Although TMAC improves on SMAC's energy savings, it still suffers from the main problem of the high cost of maintaining common active/sleep schedules via exchanging *SYNC* messages.

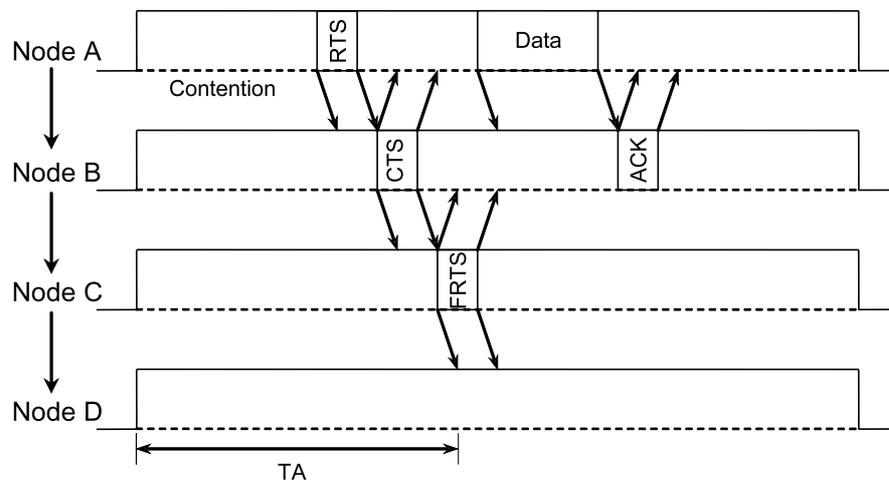


Figure 2.3: The transmission of FRTS aims at keeping node D awake. FRTS control frames make it possible for TMAC to achieve a transmission over three hops within a single active period. This technique reduces the end-to-end latency.

Minimizing Sleep Delay:

The adaptive listening technique proposed in [44] suggests the use of overhearing to reduce the sleep delay. In adaptive listening, the node that overhears its neighbor's transmission and learns from it when that transmission ends may sleep in the meantime and then wakes up just when the transmission ends. The node wakes up after that transmission even if it might happen during its sleep period. This makes it possible for the node's neighbor to immediately send data to it, instead of waiting for the node's next scheduled active time.

Technically speaking, nodes can learn about when a transmission ends if they receive the frame header that indicates the frame length or if they receive RTS or CTS that precede the data frame — RTS and CTS frames indicate the transmission duration of the data frame.

In slotted protocols that use common active/sleep schedules, the sleep delay is a serious drawback because it increases communication latency. Improvements such as adaptive listening [44] and TMAC [42] only affect the next hop and the next two-hop nodes respectively. The following protocols aim at minimizing the sleep delay furthermore.

The DSMAC (Dynamic SMAC) [45] protocol dynamically changes each node's duty-cycle to meet applications' new and changing demands. A node increases its duty cycle by adding extra active periods when it requires less latency or when it observes an increasing traffic load. DSMAC assume that all nodes start with the same duty cycle. Then, when a node needs to increase its duty cycle, it sends a *SYNC* message to its neighbors to inform them about its additional active schedule. After receiving the *SYNC* message, each neighbor locally decides whether to increase its duty cycle to meet the announced schedule or not. The key idea that makes DSMAC work even

with nodes that do not increase their schedules is that, initially, active periods never get changed; nodes only insert their new active schedules in the middle of the sleep period. Note that nodes can also decrease their duty-cycles by removing the added active periods.

In FPA (Fast Path Algorithm) [34], nodes wake up for an additional time, even during their pre-scheduled sleep periods, to ensure timely relaying of frames. A node uses its hop-distance from the sender to estimate when its upstream neighbor will send a frame to it. Then, the node wakes up at the estimated time only to receive and potentially forward the frame to its downstream neighbor. The node sets these additional wakeup times from information piggybacked in the first data message on that path.

The DMAC (Data-gathering MAC) [46] considers the situation where many sources send data to a sink through a unidirectional tree, called convergecast communication. Nodes exploit this tree to determine their active schedules. A node determines its active schedules according to the traffic load and to its depth in the tree. The active periods of DMAC are similar to the additional active periods of FPA. DMAC mainly targets stationary networks as it does not envisage common global active periods. Thus, dynamicity may decrease DMAC's performance drastically.

Handling Mobility:

The initial SMAC protocol targets stationary sensor networks and it does not envisage specific optimizations to handle nodes mobility efficiently. With mobility, the node's schedule is no longer valid whenever the node moves to another virtual cluster. To re-establishes a new schedule, a node keeps continuously listening for a duration of an active period plus a sleep period to receive a sync. Mobility decreases SMAC energy saving rates as mobile nodes waste extra energy in establishing new schedules. Furthermore, mobility increases communication latency as mobile nodes spend additional time to establish a new schedule and setup a connection.

The MSMAC (Mobility-aware SMAC) [35] proposes a mechanism that adapts the duty cycle of SMAC to improve connection setup times in mobile environments. Nodes measure changes in received signal levels of the periodic *SYNC* messages and use them to estimate mobility speed. A node first estimates mobility speed with each one of its neighbors. Then it informs its neighbors about the maximum estimated speed by including that speed in *SYNC* messages. When the node's neighbors receive the *SYNC* messages, they create an *active zone* around it. In active zones, nodes increase their active periods by staying awake longer to reduce the connection setup time.

Minimizing Schedules Number:

Multiple active schedules lower SMAC's energy saving rates as nodes spend more time in active periods. Experiments with motes, reported in [34, 47], show that more than half of the nodes have more than one active schedule. The GSA (Global Schedule Algorithm) [34] focuses on minimizing the number of active schedules by making all nodes within a sensor network converge to a common global schedule. The GSA uses the schedule's age to determine which schedule to keep; when a node has to select between many schedules, it selects the oldest one. Results reported in [34] show that

the GSA converges to one schedule in a network of 40 nodes organized in a linear topology.

2.2.2 Preamble Sampling

Basic Idea

Preamble sampling protocols do not use common active/sleep schedules; instead, they let each node choose its active schedule independently of other nodes around. In preamble sampling protocols, a node spends most of the time in sleep mode; it wakes up only for a short duration to check whether there is a transmission on the channel. To avoid deafness, each data frame is preceded by a preamble that is long enough to make sure that all potential receivers detect the preamble and then get the data frame.

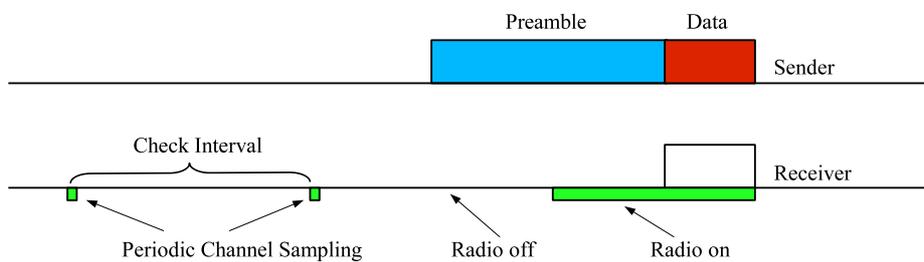


Figure 2.4: *Preamble sampling technique*

Figure 2.4 shows an example of a preamble sampling protocol operation. According to the duty-cycle parameter, nodes periodically switch their radios on to sample the channel. If a node finds that the channel is idle, it goes back to sleep immediately. However, if it detects a preamble transmission on the channel, then it keeps its radio on until it receives the subsequent data frame. Right after the reception of the data frame, the node sends an ACK frame, if needed, and goes back to sleep afterward. To be effective, the duration of the preamble transmission needs to be at least as long as the *Check Interval (CI)* defined as the period between two consecutive instants of node wakeup. In this way, a node makes sure that all potential receivers are awake during its preamble transmission so that they get the subsequent data frame. The preamble sampling technique has been combined with Aloha in [48] and with CSMA in [49].

We can find in the literature other terminologies that refer to a similar approach, e.g. Cycled Receiver [50], LPL (Low Power Listening) [41] and Channel Polling [24]. Hereafter, these protocols are collectively referred to as preamble sampling protocols.

Discussion

By reducing synchronization overhead, preamble sampling protocols realize larger energy savings; however, this comes at the cost of a longer preamble. The use of a longer preamble causes two major problems that are:

Costly Collisions:

The preamble sampling technique shifts the cost of coping with idle listening from the receiver to the transmitter. The receiver uses less energy as it wakes up only for a very short time, whereas the transmitter uses more energy as it transmits a long preamble before each data frame. This is highly beneficial for applications in which transmission is not frequent, such as surveillance.

The high transmission cost counteracts the energy efficiency of preamble sampling protocols in situations with high collision rates. When a collision occurs, it very likely implies retransmission, which is costly. Preamble sampling protocols should have robust mechanisms for avoiding collisions and extremely low data rates.

Limited Duty Cycle:

In order to extend nodes lifetime, applications need to save more energy by lowering the duty cycle. Lowering the duty cycle implies putting nodes in sleep mode for larger periods, which means extending the check interval.

While using a larger check interval reduces the cost of idle listening at the receiver, it increases the transmission cost as the transmitter uses a longer preamble. Thus, nodes cannot indefinitely extend their check intervals with the aim of saving more energy.

There is an optimal value for the check interval beyond which nodes waste more energy in transmission than they save in reception. Finding the optimal check interval depends upon several parameters such as transmission power, reception power, traffic load and switching times of the radio chip. Figure 2.5 shows that there is an optimal value for the check interval that maximizes the lifetime. This values also depends on the traffic load. Therefore, preamble sampling protocols have a limited duty cycle that is determined by the optimal check interval value.

Further Improvements

Improving Clear Channel Assessment:

Clear Channel Assessment (CCA) is the operation that determines whether the channel is clear. In CSMA (Carrier Sense Multiple Access), a node performs a CCA before transmitting a frame to avoid causing a collision in case it transmits while the channel is busy. A common method used to performing a CCA is thresholding. As used in the IEEE 802.15.4 [38], thresholding consists in measuring the power of a received signal and comparing it to the noise floor. The channel is considered clear only if the measured signal is below the noise floor. Thresholding generates a large number of false positive because of the significant variance in channel energy [41]. False positives lower the effective channel bandwidth, thus they should be reduced.

Instead of thresholding, the BMAC [41] protocol proposes a technique based on outlier detection to improve the quality of CCA. In this technique, a node searches for outliers in the received signal such that the channel energy is significantly below the noise floor. If the node detects an outlier during channel sampling, then it declares the channel is clear because a valid signal has outliers significantly below the noise floor with low probability only. If the node does not find any outlier within fives samples, then it declares the channel to be busy. Outlier detection substantially outperforms thresholding as reported in [41].

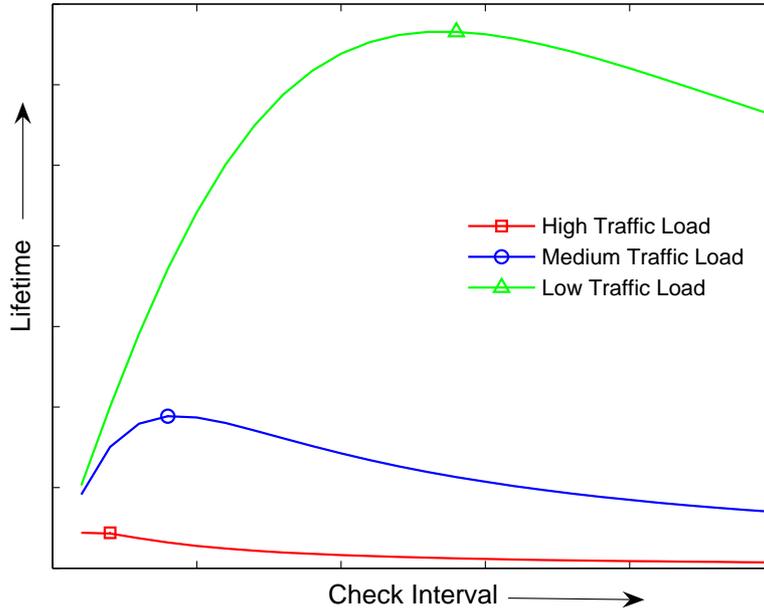


Figure 2.5: Lifetime of nodes according to different check intervals and traffic loads.

The outlier detection technique depends upon the accuracy of the noise floor estimation. BMAC uses software automatic gain control for estimating the noise floor to adapt to ambient noise changes. Each node takes signal strength samples at times when the channel is supposed to be clear, such as immediately after transmitting a frame. From these values, each node calculates an average value and uses it as a simple low pass filter for the noise floor estimate.

Apart from collision avoidance and good channel utilization, accurate CCA has additional benefits. It makes it possible for a node, listening to the preamble while waiting to receive the data frame, to determine whether the channel is still busy. In the case the node detects that the channel is back to idle before it receives the data, it stops listening and goes back sleeping. By avoiding this reception, an accurate CCA further improves preamble sampling performance.

Adapting Duty Cycle:

Determining the optimal check interval in preamble sampling protocols requires knowing applications' traffic load *a priori* because nodes have no means for adapting their check interval to traffic load changes. This constraint makes preamble sampling inflexible for applications with highly fluctuating traffic loads. BMAC [41] proposes to alleviate such rigidity through the use of a versatile low power listening in which each node has an interface for dynamically configuring several MAC layer parameters, such as the check interval. BMAC proposes eight standard listening modes corresponding to eight different check intervals. A node can dynamically switch from one listening mode to another to meet applications' new and changing demands.

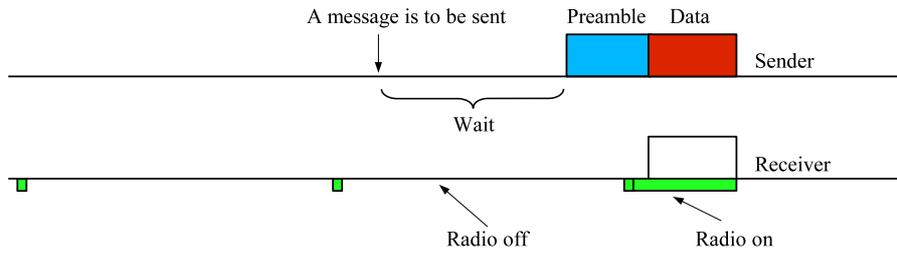


Figure 2.6: *WiseMAC improves on preamble sampling energy savings through the use of short preambles. In WiseMAC, each transmitter knowing the wakeup time of the receiver sends its transmission just on time to meet the receiver wakeup. The transmitter may use a short preamble to cover clock drifts about the receiver’s wakeup time.*

The EA-ALPL (Energy-Aware Adaptive Low Power Listening) protocol [51] exploits BMAC’s reconfiguration interfaces to adapt to traffic changes; that is, each node sets its listening mode according to its current and past forwarding loads. EA-ALPL also makes use of these different listening modes to influence routing decisions. For example, an overused node that does not want to forward other nodes’ traffic anymore, voluntarily increases its listening mode to encourage its neighbors to choose another node to continue relaying their traffic. In energy-efficient routing, a node selects the next hop with the minimum check interval because transmitting to that node consumes less energy as it requires a shorter preamble.

Cutting Back Preamble Length:

Large preambles mitigate the performance of preamble sampling protocols because nodes drain significant energy in transmission. WiseMAC (Wireless Sensor MAC) [25] alleviates this drawback by making it possible for nodes to use short preambles for some unicast transmissions.

In preamble sampling, a node that wants to transmit a data frame uses a preamble that is as long as the check interval, hereafter referred to as full length preamble. The node uses a full length preamble because it does not know when the receiver wakes up. To save the transmitter the overhead of using a full-length preamble, WiseMAC aims at letting each node learn about its neighbors’ wakeup times; if the transmitter knows the wakeup time of the receiver, then it can timely start its transmission just to meet the receiver wakeup. Clock drifts may make the transmitter lose accuracy about the receiver wakeup time. In such a case, the transmitter uses a preamble that is just long enough to make up for the estimated maximum clock drift. The length of the preamble used in this case depends on clock drifts: the smaller the clock drift, the shorter the preamble the transmitter has to use. Figure 2.6 shows an example of short preamble utilization.

Each node running WiseMAC makes use of an internal table to store its neighbors’ wakeup times. To keep maintenance and construction of such tables low-cost, nodes adopt a passive approach: each node declares its wakeup time by piggybacking it on the ACK frames used to acknowledge a successful reception. When a node receives an ACK frame, it updates its table with the wakeup time of the node that transmitted

the ACK. Note that a node may have no information about the wakeup time of a neighbor to which it wants to transmit a frame; in that case, the node uses a full-length preamble.

The SCP (Scheduled Channel Polling) [24] uses the similar idea of letting each node know its neighbors' wakeup times to lower the preamble length. However, SCP does not restrict nodes to only adopt a passive approach for the construction and maintenance of their tables. Instead, a node executing SCP can also declare its schedule to its neighbors in an active way: each node broadcasts its wakeup time every synchronization period so that its neighbors update their tables with recent values, thereby lowering clock drifts. SCP determines the optimal synchronization period that achieves the best trade-off between: actively declaring wakeup times through dedicated periodic broadcasts, and transmitting long preambles. Results reported in [24] show that SCP reduces the duty cycle of preamble sampling from 1-2% to less than 0.1%.

Using Two Separate Channels, Data and Wakeup:

The STEM (Sparse Topology and Energy Management) protocol [28] makes use of two channels: a wakeup channel and a data channel. The wakeup channel is used to organize a meeting between the transmitter and the receiver to avoid deafness, whereas the data channel is only used for data exchange once the meeting occurs. To ensure a meeting between the transmitter and the receiver, nodes follow a preamble sampling approach: the receiver periodically samples the wakeup channel and the transmitter sends preambles on the wakeup channel before sending the data on the data channel.

STEM has two preamble variants: STEM-T (STEM Tone) and STEM-B (STEM Beacon). In STEM-T, the preamble consists of a simple busy tone. Thus STEM-T is very similar to traditional preamble sampling protocols except for using two separate channels instead of only one. In STEM-B, the preamble consists of a series of beacons, each beacon carrying the MAC addresses of the transmitter and of the receiver. The node that wakes up to sample the channel expects to receive a beacon instead of finding a tone. When the node receives a beacon, it learns from the MAC addresses carried therein whether it is the destination of the data frame. If so, the node sends an acknowledgment back to the sender (note that beacons are not transmitted contiguously), inter beacon blanks being intentionally inserted to let the receiver send its acknowledgment. When the transmitter receives an acknowledgment, it stops transmitting beacons and switches to the data channel to send the data frame. After sending the acknowledgment, the receiver also switches to the data channel to receive the data.

STEM-B has the advantage of cutting back the preamble length as transmitters do not require to always use a full-length preamble. However, STEM-T uses a simpler transceiver on the wakeup channel, which can be significantly cheaper and less energy consuming than a transceiver used for data communication.

Many other protocols, such as CSMA-MPS (CSMA with Minimum Preamble Sampling) [27], TICER (Transmitted Initiated Cycled Receiver) [50], WOR (Wake On Radio) [11] and X-MAC [26], use techniques similar to STEM-B, but with a single channel: beacons are transmitted on the same data channel.

Protocols that use preambles split into frames with a gap between consecutive frames have the advantage of not always requiring the full-length preamble; in the case of unicast transmissions, the receiver sends the ACK in the gap between the frames, thus stopping the preamble transmission. However, in very lightly loaded networks, these protocols do not guarantee optimal energy savings, because they increase idle listening at the receivers. When there is a gap between frames, nodes should stay in receive mode for a duration larger than the gap to sample the channel. Therefore, the sampling duration increases and thus nodes waste more energy in sampling.

Initiating Communication by the Receiver:

The RICER (Receiver Initiated Cycled Receiver) [50] shifts communication initiation from the transmitter side to the receiver side. When the receiver wants to receive a frame, it sends a beacon to announce that it is awoken. Right after beacon transmission, the receiver monitors the channel for a certain time waiting for a response from the sender. If the receiver gets a response, it transmits the data just afterward, otherwise, it goes back to sleep. To send a data frame, the transmitter stays awake and monitors the channel waiting for a beacon from the receiver. Once the transmitter receives the beacon, it transmits the data frame and waits for an ACK to end the session. RICER achieves high energy saving for unicast and anycast communications. However, it cannot be used for broadcast and multicast communications, because it is receiver-initiated.

The idea of RICER is similar to preamble sampling; however, the transmitter keeps receiving instead of transmitting a full-length preamble. The receiver periodically sends frames to announce it is ready to receive frames and monitors the channel thereafter to receive the ACK and the transmission. This overhead is large in lightly loaded networks as the receiver does it periodically.

Mixing Preamble Sampling with TDMA:

The Z-MAC (Zebra MAC) [52] protocol addresses the weaknesses of CSMA schemes such as collisions and bad channel utilization under high contention. Z-MAC is a hybrid protocol that combines CSMA and TDMA schemes to take advantage of their strengths. Under low contention, Z-MAC switches to a CSMA to achieve high channel utilization and low delays. Under high contention, Z-MAC switches to TDMA to achieve high channel utilization, fairness, and less collisions. Z-MAC uses DRAND [53], a distributed solution for TDMA slot distribution among nodes. In contrast to traditional TDMA schemes, a node can transmit in both its own time slot and in other slots assigned to other nodes. However, the owner of a slot always has high priority over the others; nodes that want to transmit in others' time slots use a random backoff for contention within the others' time slots. In this way, Z-MAC switches to CSMA as nodes having traffic to send utilize TDMA slots that are not used by their owners.

2.3 Standardization

Although there are a lot of energy-aware MAC proposals, there is still no standard for low power multihop wireless sensor networks. Recently, the IEEE has standardized 802.15.4 [38]

for low data rate and low power area networks. The IEEE 802.15.4 standard may be inefficient in node-homogeneous networks as it is mainly designed to target Zigbee [54] applications. To meet the needs of Zigbee, the IEEE 802.15.4 defines two type of nodes: FFD (Full Functional Devices) acting as Coordinators (PAN Coordinators or Routers) and RFD (Reduced Functional Devices) acting as End Devices. From Zigbee applications point of view, only RFDs do need to save energy since FFDs are assumed to be continuously powered. Although the IEEE 802.15.4 standard envisages a beacon mode in which there are active and sleep periods, it is not clear how to make FFDs (internal nodes of the network) take advantage of this feature, neither in mesh nor in tree network organizations. The main problem is how to efficiently distribute active/sleep schedules among nodes so that there are no collisions during beacon transmissions. Another weakness of the IEEE 802.15.4 arises with variable traffic loads when the traffic load falls. With low traffic rates, the overhead of systematically sending and listening to beacons to maintain synchronization, may be very costly.

2.4 Conclusions

Idle listening is the main source of energy waste in sensor networks because without special management nodes spend large amounts of time listening to an idle channel. Therefore, the main challenge of a low power MAC protocol is to minimize idle listening while sustaining connectivity between neighbor nodes. Minimizing idle listening implies putting nodes in sleep mode whenever possible, i.e. when they have no frames to transmit or to receive. Sustaining connectivity implies that nodes should not miss frames transmitted to them when they sleep.

Throughout this chapter, we have dwelt on two major techniques that lower idle listening while sustaining connectivity. We have discussed the drawbacks and advantages of each, namely common active/sleep schedules and preamble sampling techniques. We have shown that using common active/sleep schedules in networks with time-varying and irregular traffic loads is unsuitable because the lengths of sleep and active periods are calculated according to a predefined traffic load. When the traffic load increases, the active periods may not be large enough to absorb all the traffic, which results in increased contention and more collisions. However, when the traffic load decreases, the active periods may be too large and then nodes waste more energy in idle listening. Moreover, using common sleep/active schedules requires exchanging synchronization messages for the construction and maintenance of the schedules, which represents a non negligible overhead when the traffic load decreases. Results reported in [55] also confirm that preamble sampling protocols outperform common active/sleep schedule protocols. These results guided our research toward focusing on preamble sampling protocols (see Chapter 6 and Chapter 7).

The MAC layer provides the most important functionalities that should be investigated to increase the lifetime of a single node, however, it is not sufficient alone for extending the lifetime of the whole network. Extending the network lifetime time requires taking into account the roles of all the nodes in the network; for example, nodes that experience high traffic loads will run out of energy earlier causing a possible crash of the application. As the amount of traffic each node forwards depends on the used routing protocol, network lifetime extension requires an optimization at the routing layer in addition to that done at

the MAC layer. We hence devote the next chapter to the state of the art on optimized routing protocols for sensor networks.

2 *Channel Access Protocols*

3 Routing Protocols

The role of a routing protocol in a wireless multihop network is to ensure communication between nodes that are not in the transmission range of each other by relaying packets through intermediate nodes. The first routing protocols mainly designed for MANETs (mobile ad hoc networks) in which bandwidth, delay, and mobility are the main concerns [56, 57], may be inefficient for sensor networks as they have not been optimized for such tightly resources-constrained networks. This chapter presents a quick¹ survey on routing in wireless sensor networks. We start by listing the major characteristics of sensor networks that influence the design of a routing protocol, namely the data-centric communication scheme and the scarce energy and computation resources. We then describe two major classes of routing protocols: data-centric routing and energy-efficient routing. These two approaches can be combined so that the resulting routing protocols take advantage of both their qualities, as shown in Chapter 8.

3.1 Characteristics of Routing in WSN

3.1.1 Data-Centric Communication

Sensor networks do not require traditional node-centric communication scheme in which messages are relayed according to their destination addresses—both nodes interested in gathering information from the network and those providing information to the network may not know a priori the identities of nodes to which they send requests and data messages. Thus, a data-centric communication scheme in which nodes address data-contents instead of node-identifiers is much more suitable.

Data-centric communication offers the following benefits. It works in networks with duplicate node-identifiers, which saves the overhead of maintaining unique node-identifiers. It also makes data aggregation possible, which saves the overhead of transmitting redundant messages. The most popular routing protocol using a data-centric communication scheme is Directed Diffusion [61]. We devote Section 3.2 to describe Directed Diffusion and its main variants.

3.1.2 Scarce Energy and Computation Resources

Sensor networks have finite energy reserves and very low processing and storage capabilities. A well designed routing protocol should be energy-efficient, have low computation complexity, and select reliable routes. The main goal of energy-efficiency is to maximize the lifetime

¹The literature on routing in WSN is very rich [58, 59, 60]. Therefore, an exhaustive description of all protocols is beyond the scope of this chapter. We just discuss the major contributions related to our work presented in Chapter 8

3 Routing Protocols

of sensor networks. Energy-efficient routing protocols rely on the following techniques: reducing the overhead of the protocol, mainly the number of exchanged routing messages, and selecting good routes that reduce the energy consumed per transmitted packet while avoiding overusing vulnerable nodes. In Section 3.3, we present some examples of protocols that use these techniques.

3.1.3 Other Features

Sensor networks may have other characteristics that influence the design of the routing protocol. For instance, sensor networks are assumed to exhibit only weak dynamicity that essentially stems from time-varying radio conditions, newly added, recently dead, or sleeping nodes. This weak mobility may change the design of a routing protocol (e.g. use a reactive approach rather than a proactive one).

Another characteristic that further simplifies the design and improves the performance of the routing protocol is the availability of nodes' positions [62, 63, 64, 65]. Although such an information is highly useful, we choose not to rely on it in the work presented in this dissertation for the following reasons. First, we want to cover a broader range of networks and applications in which acquiring nodes positions is not possible. Second, we argue that getting geographic positions may be inappropriate for some applications: either nodes have small and low power GPS devices, which is costly, or they execute GPS-free positioning algorithm [66, 67, 68] to calculate their positions, which generates a high overhead.

3.2 Data-Centric Routing

In data-centric routing, nodes usually use a publish/subscribe approach in which receivers (sinks) subscribe to information that is of interest to them without regard to any specific source and senders (sources) publish the information they acquire without addressing it to any specific destination. In the next section, we present Directed Diffusion [61], one of the most popular implementations of the publish/subscribe approach.

3.2.1 Directed Diffusion

To perform a data-centric routing, directed diffusion uses an attribute-value naming scheme to make messages' contents accessible to intermediate nodes. This naming scheme can also be seen as a powerful interface between the routing protocol and applications. Relaying messages in Directed Diffusion is based on several elements such as interests, data, gradients, and reinforcements. An interest is a query that specifies what a user wants (e.g., the temperature of a specific region R). Data are results of what sensors sense: a data message may be an answer to a given query expressed in interests (e.g., the temperature of region R is 25). Gradients are pointers to downstream nodes from sources to sinks for a given interest. Reinforcement is the process of selecting the best gradients to form a path along which sources send data to sinks. These elements are described in details in the following sections.

Attribute-Value Naming

In directed diffusion, all the relayed messages are expressed in attribute-value pairs. For example, a node tracking a wheeled vehicle within a particular region and seeking results each 20ms for a duration of 10s, broadcasts an interest message with the following format:

```

type          = wheeled vehicle // detect an object of type wheeled vehicle
interval     = 20 ms // send events (data) every 20 ms.
duration     = 10 s // for the next 10 s
rectangular  = [-100, 100, 200, 400] // the coordinates of the region of interest.

```

Any node within the considered region that receives the interest and detects a wheeled vehicle sends data messages back to the node that issued the interest message. Similarly, the data message should also follow the attribute-value naming scheme. For example, a data message may look like the following:

```

instance     = truck // instance of the wheeled vehicle.
location    = [125, 220] // location of the node that issues this data.
intensity   = 0.6 // signal amplitude measure.
confidence  = 0.85 // confidence in match.
timestamp   = 01: 20: 40 // event generation time.

```

More details on the naming scheme used in Directed Diffusion can be found in [69, 70].

Interest Propagation

Interests are task descriptions expressed in the attribute-value naming scheme and injected into the network by sink nodes. To propagate an interest throughout the network, a sink node starts by broadcasting it to its neighbors. Then each neighbor that receives the interest creates a corresponding routing entry (or updates an existing one). This routing entry contains a list of gradients, which are pointers to the nodes from which the neighbor has received the interest. Likewise, neighbors of neighbors continue propagating the interest until gradients are installed throughout the networks. Figures Figure 3.1 and Figure 3.2 show an example with one source and one sink. Note that gradients are soft-state pointers that expire after a certain timeout value, thus sink nodes should periodically broadcast interests to refresh the existing gradients and potentially install new ones.

Gradient Reinforcement

The reinforcement phase consists in selecting some good gradients to build routes from sources to sinks. The selected gradients are called reinforced gradients. The algorithm used for reinforcing gradients depends upon the used variant of Directed Diffusion: one-phase pull, two-phase pull, and push. We present the details of the gradient reinforcement algorithm for each variant in Section 3.2.2.

Data Transmission

Each node that receives an interest message and is able to produce data becomes a source node. Each source node constructs data packets and sends them back to the sink node(s) according to the gradients installed during the interest propagation phase. Data messages

3 Routing Protocols

protocol	sink	source
two-phase pull	(1) interest (flooded)	
		(2) exploratory data (flooded)
	(3) positive reinforcement	(4) data
one-phase- pull	(1) interest (flooded)	(2) data
push		(1) exploratory data (flooded)
	(2) positive reinforcement	(3) data

Table 3.1: Summary of the three variants of Directed Diffusion. Numbers indicate the sequence of operations

try to follow in priority the reinforced gradients. Otherwise, they follow simple gradients. Note that routing tables are different in Directed Diffusion; a routing entry consists of an interest description and a list of gradients pointing to the next downstream nodes to the sink(s). It does not have any information about the number of sinks interested in the data nor about their identities.

Data Aggregation

The use of the attribute-value naming scheme makes it possible for nodes to perform some in-network processing operations such as data aggregation, thus improving the performance of Directed Diffusion. For example, assume that a node has received an interest I_1 , created a routing entry for that interest, and installed the corresponding gradient. When this node receives another interest I_2 with a similar task description, it does not create another entry, it only updates the existing routing entry with the new potential gradient, which reduces the number of entries in the routing table. The data aggregation mechanism also applies to data messages. When a node receives multiple similar data messages, it performs aggregation and sends only one copy of the data, which reduces the traffic and thus saves energy.

3.2.2 Variants of Directed Diffusion

Directed Diffusion has three principal variants that have been optimized for a certain type of networks [71]. The principal differences between these variants are summarized in Table 3.1.

Two-Phase Pull Directed Diffusion

The two-phase pull directed diffusion, also referred to as 2PP diffusion, works as follows. First, a sink starts by broadcasting an interest message throughout the network. Then each node having data that corresponds to the interest query replies by flooding exploratory data so that they eventually reach all the potential sinks. Note that the interest propagation and the exploratory-data flooding have different goals: the interest propagation only asks the network to perform a given task, whereas the exploratory-data flooding aims at discovering routes from sources to sinks.

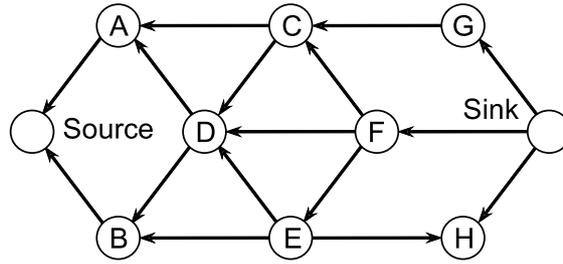


Figure 3.1: Interest propagation.

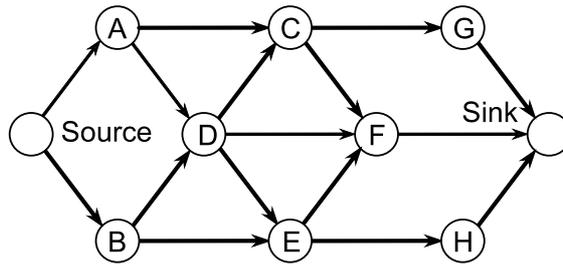


Figure 3.2: Gradient setup.

The sink selects a good route among all the routes discovered from a source node to a sink node. It transmits the data messages along this route. One of the metrics that can be used for route selection is path latency. In this case, each intermediate node marks the identity of the upstream node from which it received the first exploratory-data message. When the sink receives an exploratory data, it sends a gradient reinforcement message to the upstream neighbor from which it received the first exploratory-data message. In the example presented in Figure 3.3, the sink sends a gradient reinforcement message to Node *F*. Upon the reception of this reinforcement message, Node *F* marks its gradient toward the sink as reinforced, then it continues the gradient reinforcement phase by sending a reinforcement message to Node *D*. Meanwhile, the procedure of gradient reinforcement continues backward until a positive reinforcement message reaches the source. When the source receives a positive reinforcement (from Node *A* in the example) it reinforces the gradient toward Node *A* and starts transmitting data messages that follow the route indicated by the reinforced gradients (Figure 3.4) hop-by-hop until they reach the sink.

By using two phases (exploratory-data and gradient reinforcements) for route construction, the 2PP diffusion aims at reducing the number of transmitted data messages in networks with duplicate node-identifiers. Figure 3.5 shows an example in which a traditional routing protocol that finds routes in function of the destination node identifier generates extra data messages. The example of Figure 3.5 shows two cases: a network with two different sinks with the same identifier (Network 1) and another network with only one sink (Network 2). With a traditional routing, Node *A* does not know whether it is in the situation presented by Network 1 or in that presented by Network 2. Therefore, Node *A* should send data messages to both nodes *B* and *C* to guarantee message delivery in the worst case (Network 1), which is not optimal for the Network 2 as only one message to either *B* or *C*

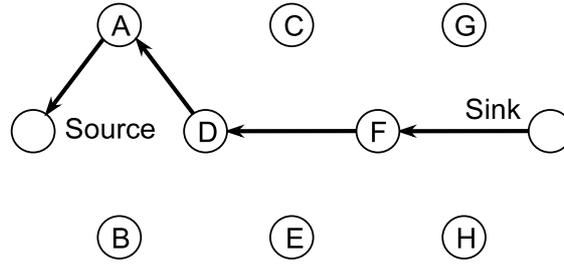


Figure 3.3: Gradient reinforcement.

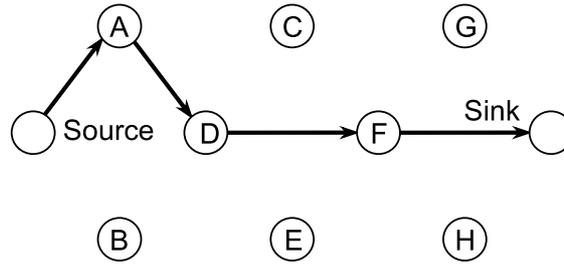


Figure 3.4: Data path.

is enough. This shortcoming is avoided by 2PP as node A exactly sends the right number of messages, i.e. two data messages to both B and C , in case of Network 1, and one data message to either B or C , in case of Network 2.

In the case of Network 1, when exploratory data reach the sinks, Sink D (resp. D') sends a reinforcement message asking node B (resp. C) to reinforce the gradient $B \rightarrow D$ (resp. $C \rightarrow D'$). Next, Nodes B and C do the same and ask Node A to reinforce gradients $A \rightarrow B$ and $A \rightarrow C$. Then, Node A asks the source to reinforce gradient $S \rightarrow A$. After these reinforcements, Source S start sending data packets to Node A that in turn forwards them to Sinks D and D' through Nodes B and C respectively.

However, in the case of Network 2, Sink D'' only asks the node from which it first received an exploratory data packet to reinforce its gradient, assume it is Node B . In this case, Sink D'' asks node B to reinforce the gradient $B \rightarrow D''$. Next, Node B asks Node A to reinforce gradient $A \rightarrow B$. Then, Node A does the same and asks Source S to reinforce gradient $S \rightarrow A$. After these reinforcement, Source S start sending data to Sink D'' through the route $S - A - B - D''$.

Push Diffusion

The Push Diffusion [71] is designed for applications in which the frequency of communication is very rare. The Push Diffusion is similar to the 2PP diffusion except that it omits the interest propagation phase to save the nodes the overhead of maintaining fresh gradients while nothing is detected. The Push Diffusion performs well in applications with many sources producing data occasionally. However, it is not a good choice for applications in which sources produce data frequently as this may result in sources sending data even when not needed.

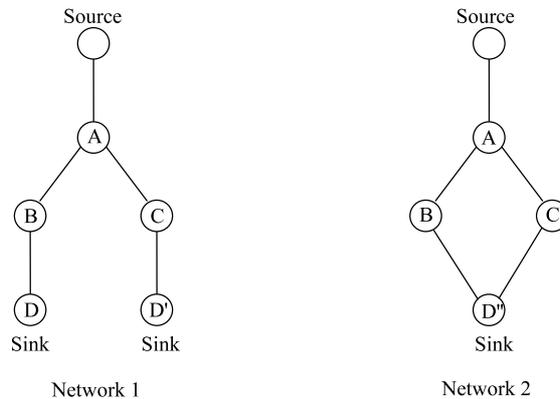


Figure 3.5: An example showing how diffusion tolerates duplicate node identifiers.

One-Phase Pull Directed Diffusion

The One-Phase Pull Directed Diffusion [71], also referred to as 1PP diffusion, is designed to lighten the overhead of 2PP diffusion by getting rid of exploratory data and reinforcement phases. In contrast to the other variants, the 1PP diffusion installs and reinforces gradients at the same time, during the interest propagation phase. Each node that receives an interest sets up a reinforced gradient toward the downstream neighbors from which it first receives the interest. Thus, once an interest reaches a source, it starts transmitting data packets to the sink according to the just installed and reinforced gradients.

The 1PP Diffusion maintains the feature of tolerating duplicate node-identifiers by using randomized flow-identifiers. The hope is that two sinks having the same node-identifiers and sending similar interests will pick different random flow-identifiers so that they can be distinguished. This technique is to be used only when there are few sinks so as to keep the collision probability of flow identifiers low.

3.2.3 Discussion

Directed Diffusion is an interesting routing and communication paradigm well designed to work in sensor networks. Its routing part allows duplicate node-identifiers that are expected to occur in largely deployed sensor networks. Its communication schemes based on attribute-value naming scheme provides a powerful interface to applications and makes in-network processing possible, which saves more energy through data aggregation. Numerous works have been proposed to improve Directed Diffusion. For instance, [72] and [73] propose to balance the load of relaying data messages by using a multipath routing between sources and sinks. Another improvement in [74] proposes to use geographic information to restrict the propagation of interest according to the coordinates potentially expressed in the query. Many other improvements consider an explicit energy metric when finding routes. We discuss these energy-efficient techniques in the next section.

3.3 Energy-Efficient Routing

In an energy-efficient routing, nodes use one or a combination of the following strategies: (i) reducing protocol overhead by minimizing the number of messages exchanged unnecessarily, and (ii) selecting good routes by using explicit energy-related metrics. These strategies aim at optimizing energy utilization, which increases the lifetime of the network. In Chapter 8, we show that these strategies can be used together within a single routing protocol that reduces the number of exchanged routing messages while selecting good routes. In the following sections, we illustrate these strategies through examples of routing protocols.

3.3.1 Reducing Protocols Overhead

This strategy consists in forwarding packets without using routing tables. This usually happens when routing tables are not constructed yet or when the overhead of creating and maintaining these tables is considered excessive. Avoiding such overhead contributes to more energy saving especially in some types of applications such as those in which a node issues a command and does not expect any answer.

The Flooding is the simplest and most robust mode of forwarding without routing tables. In flooding, a node broadcasts each new message it receives. However, flooding may be costly in terms of energy and bandwidth as many routing messages are propagated unnecessarily—unnecessary messages are those transmitted by nodes with no additional coverage area. In the following sections, we present some solutions that have been proposed as alternatives to flooding; we take as examples the gossip-based routing, the optimized flooding, and the rumor routing.

Gossip Based Routing

The Gossip based approach proposed in [13] aims at reducing the amount of unnecessarily propagated messages of a routing protocol while ensuring the coverage of most of the nodes. In contrast to flooding, a node using a gossip based routing forwards each new message it receives according to a predefined gossiping probability. Results reported in [13] show that gossip exhibits a bimodal behavior in case of constant gossiping probability. When this probability is below a certain threshold, gossip based routing dies out quickly and reaches a small number of nodes. However, when the gossiping probability is larger than the threshold, gossip based routing reaches almost all the nodes. Although the value of the threshold depends upon the network topology, [13] reports that it is generally between 0.6 and 0.8. Simulation results show that gossip generates up to 35% less messages than flooding does. This reduction of the number of messages is assumed to reduce the energy consumption of nodes. However, as gossip also finds routes that are 10-15% larger than those flooding finds, the amount of the saved energy is reduced.

Optimized Flooding

The main drawbacks of flooding are: (i) redundancy, caused by forwarding the same message by all the nodes and (ii) collisions, caused by CSMA contentions. The paper [14] studied these problems and proposed many optimized flooding schemes to alleviate them. The proposed schemes (e.g. probabilistic, counter-based, distance-based, and location-based)

depend on how a node estimates the redundancy and on the accuracy of the knowledge it accumulated to assist its decision.

The probabilistic scheme is similar to the gossip based routing. In this scheme, a node broadcasts each new message it receives with a constant probability p . When p is equal to 1 the probabilistic scheme is equivalent to flooding. Note that each broadcast is preceded by a random delay in order to alleviate the contention among node and thus reduce collisions.

The counter-based scheme exploits the fact that a node may receive more than one copy of the same broadcast before it sends its own one. In this approach, a node defines a certain threshold C , which is the number of received redundant copies beyond which the node decides to cancel message forwarding. The rationale behind this technique is the following: the more neighbors of a node forward, the less the expected additional coverage ensured by the forwarding of the node.

The distance-based scheme makes use of the relative distance between the node and its neighbors from which it has received broadcast messages. It defines a distance d_{min} , which is the distance to the closest neighbor from which it has received a broadcast. The scheme also defines a threshold distance D . The node broadcasts only when d_{min} is larger than D . The rationale behind of this scheme is the following: the larger the distance between nodes the larger the additional coverage by the node's broadcast.

All these schemes do reduce the number of transmitted messages. However, as they are based on heuristics, the full coverage of the networks can not be guaranteed. The coverage issue can be solved by other schemes assuming that nodes are aware of their geographic positions, e.g. the location-aided [14] and the angle-based [15] schemes.

Rumor Routing

The Rumor routing [16] mainly concerns event notification applications in which some sensors are interested in certain events (e.g., a temperature exceeding a certain value). Instead of flooding the query or flooding the event notification, the rumor routing installs two types of paths: the query path and the event notification path as shown in Figure 3.6. The motivation is that the query path intersects with the event notification path with high probability. When this intersection happens, a route is created between the source (sending the event notification) and the sink (sending the query).

Results reported in [16] show that the probability of intersection is 69% when using one event path and one query path and is 99.7% when there are five query paths and five event paths. The rumor routing has the same advantages and drawbacks as the gossip based routing. As an advantage, it saves more energy by reducing the number of transmitted messages compared to the flooding. As a drawback, the transmission of messages may drain more energy as the selected routes are not guaranteed to consume the minimum of energy.

3.3.2 Using Explicit Energy-Related Metrics

This strategy consists in selecting good routes to maximize the lifetime of a sensor network. The routing algorithm uses explicit energy-related metrics for route selection. The metrics typically used are one or a combination of the following: (i) the drained energy on links and (ii) the residual energy of nodes. In this section, we present these metrics through examples of protocols. We present protocols that select the minimum energy routes, protocols that

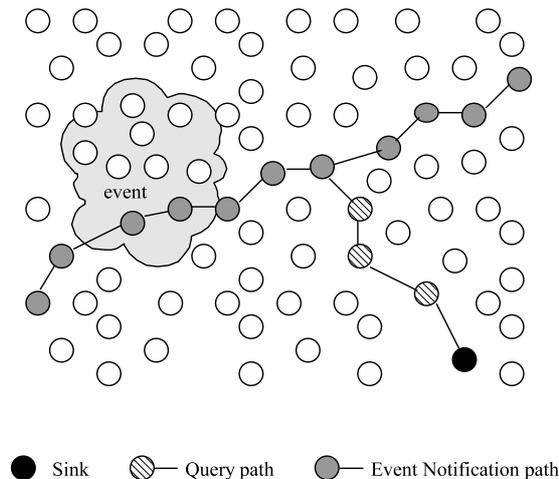


Figure 3.6: Operation of rumor routing.

avoid selecting vulnerable routes (max-min metric), and protocols that combine these two metrics.

Drained Energy on Links

In this metric (also referred to as the min metric), nodes select the route that consumes the least amount of energy. Usually, nodes adjust their transmission power and construct a minimum energy topology to reduce the overall energy consumption of the network (e.g. [18, 75, 76, 77, 78]). The resulting topology guarantees that each node communicates with other nodes using the route that consumes the least amount of energy possible overall. For example, the network presented in Figure 3.7 contains three routes between the source and the sink. The cost of the route A-C-F-G is $1+1+1 = 3$, the cost of the route A-D-G is $2+2 = 4$, and the cost of the route A-B-E-G is $2+1+3 = 6$. In this situation, the minimum energy routing selects the routes with the minimum transmission cost, i.e. route A-C-F-G. Note that this routing may select vulnerable routes. The route A-C-F-G is vulnerable as it contains a vulnerable node (node C).

Residual Energy of Nodes

In this metric (also referred to as the max-min metric), protocols use the residual energy to avoid premature energy exhaustion of nodes (e.g. [79, 80, 81]). Usually, nodes estimate their residual energy and cooperate to prevent the most vulnerable ones from being overused. By bypassing vulnerable nodes, these routing protocols ensure load balancing among nodes and avoid early network fragmentation. The most prominent protocol of this class is MMBCR (Maximum Minimum Battery Capacity Routing); we take it as a representative example.

The MMBCR protocol is mainly designed to protect nodes with low remaining energy from prematurely running out of energy. The algorithm first finds the minimum energy node on each route between a source node and a sink node, then it selects the route with the largest minimum. In the example shown in Figure 3.7, the minimum energy nodes on Routes

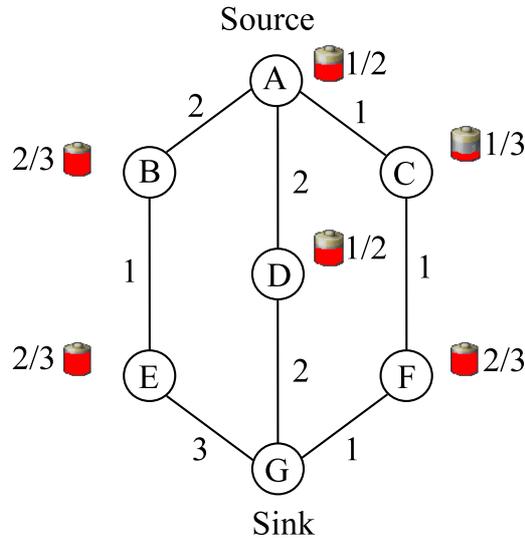


Figure 3.7: An example with nodes having different remaining energies and different link costs.

A-C-F-G, A-D-G, and A-B-E-G are Nodes B (or E), D, and C, respectively. The remaining energy of each one of these nodes defines the cost of the route to which it belongs. As the algorithm selects the routes with the largest cost, the selected routes is Route A-B-E-G. The MMBCR routing perform good load balancing between nodes, thus preventing some of them from being overused, which increases the lifetime of routes. However, the selected routes may be long, which increases the energy drained per transmitted packet. Therefore, there is a need for making a trade-off between route lengths and the remaining energy of nodes. In the next section, we present protocols that make such a trade-off.

Combining Both Metrics

Protocols of this class combine the two preceding energy metrics, namely the min metric and the max-min metric. The rationale for this combination is that these two metrics are complementary. Indeed, at the beginning of the network life, the network is dense and nodes have high residual energy so the use of a pure max-min metric may be counter effective—by trying to protect nodes with low residual energy, the max-min metric always selects routes for which the most vulnerable node has the highest residual energy; such a route may actually dissipate more energy than others. Thus, the min energy metric is a better choice when nodes have enough energy, i.e. their residual energies are larger than a predefined threshold. The max-min residual energy metric should be used to protect nodes with low residual energy, i.e. less than a predefined threshold. In the following, we present some protocols using a combination of min and max-min metrics.

CMMBCR: Conditional Maximum Minimum Battery Capacity Routing

The CMMBCR protocol [19] introduces the notion of battery protection margin γ , ($0 \leq \gamma \leq 100$). It differentiates between two kinds of routes: A and Q . Q is the set of all possible routes between a source and a sink. A , a subset of Q , is the set of the

3 Routing Protocols

routes having remaining energy greater than γ , i.e. all the nodes on each route in A have remaining energies larger than γ . The protocol is the following: when there is no route in A , the protocol selects a route in Q according to the max-min remaining energy routing MMBCR to protect the most vulnerable nodes. Otherwise, when there are routes in A , the algorithm selects the minimum energy route in A to save energy. Results reported in [19] show that CMMBCR outperforms pure min and max-min metrics, and that the performance of CMMBCR depends upon the good choice of γ .

CMRPC: Conditional Maximum Residual Packet Capacity

The CMRPC protocol [20] adds the link error rate to evaluate the energy drained on a link in the min metric. To combine the min and the max-min metrics, it defines a node-link metric, C_{ij} , for each link $i \rightarrow j$. This metric depends on the remaining energy B_i of node i , and on the transmission power ζ_{ij} needed to send a packet from i to j . Explicitly, $C_{ij} = B_i/E_{ij}$. The node-link metric determines the lifetime of the link $i \rightarrow j$. The lifetime Life_R of a route R depends on the lifetime of its most vulnerable link. Specifically, $\text{Life}_R = \min\{C_{ij}\}$, where $i \rightarrow j$ is a link on route R . The protocol executes the following algorithm: given a set of routes between a source and a destination node, choose the route with the largest lifetime.

Note that basic MRPC is a pure max-min residual energy routing, which could have undesirable behavior by always tending to protect the most vulnerable link. To cope with this issue, the CMRPC (Conditional MRPC) [20] uses a life protection threshold γ by analogy to the battery protection threshold [19]. That is, CMRPC first tries to select the route with the minimum energy consumption among the routes whose lifetimes are larger than γ . Otherwise, if there is no route satisfying this condition, CMRPC switches to MRPC. Simulation results show that CMRPC improves the performance of MRPC in terms of lifetime maximization only if the control parameter γ is well determined.

Max-Min zP_{min}

This algorithm [21] first computes P_{min} , the minimum energy needed to transmit a packet from a source to a sink across all possible routes. Then it selects a route according to the max-min residual energy as long as the energy needed to transmit a packet on selected route does not exceed $z \times P_{min}$, ($z \geq 1$). If no route satisfies these conditions, the Max-Min zP_{min} selects a route according to the min metric. Note that z is a design parameter that limits the amount of the energy drained to transmit a packet on the selected route. It prevents the max-min algorithm from selecting excessively high energy consumption routes. As the performance of the algorithm depend on the value of z , a centralized algorithm based on the gradient descent technique has been proposed to determine the optimal value of z . The Max-Min zP_{min} has a considerable overhead as it has two rounds. First it finds the energy drained on the min route, then it finds a good route according to max-min. To reduce the overhead of the algorithm, its distributed version has been proposed in [22]. However, it requires establishing synchronized mini slots at the MAC layer.

Minimum Battery Cost Routing (MBCR):

This protocol defines an energy-aware metric that considers the reluctance of a node to route traffic. The reluctance of a node increases as its battery is drained. For example, the reluctance can be defined as the reciprocal of the remaining energy [80]. The cost of a path is the sum of reluctances of nodes along this path. The routing algorithm selects the route with the smallest cost. As the reluctance metric gives high cost to nodes with low remaining energy, the MBCR algorithm steers traffic away from routes with low remaining energy nodes. In the example shown in Figure 3.7, there are three routes between the source and the sink. The cost of the route A-C-F-G is $1/(1/3) + 1/(2/3) = 4.5$. The cost of the route A-D-G is $1/(1/2) = 2$. The cost of the route A-C-F-G is $1/(2/3) + 1/(2/3) = 3$. Therefore, the selected routes is the one with the minimum cost, i.e. Route A-D-G.

MBCR implicitly takes into account a sort of trade-off between the remaining energy of nodes and routes length in terms of the number of hops.

Probabilistic Route Selection

This algorithm [23] proposes a probabilistic route selection scheme to reduce the forwarding load on minimum energy routes. The key idea of the algorithm is the following: given a set of routes between a source and a destination node, assign to each route a probability of being selected so that the minimum energy route has the highest probability. Then, forward packets on routes according to their probabilities. Note that routes with too much energy consumption, by analogy to the $\max \min zP_{min}$ algorithm [21], are assigned zero probability so that they will never be selected. Like Max-Min zP_{min} , the probabilistic route selection protocol has a considerable overhead as it needs to calculate the cost of each route between a source and a sink to assign them their corresponding route selection probabilities.

3.4 Conclusion

Sensor networks have many characteristics that make them different from traditional wireless multihop networks such as MANETs. These characteristics, mainly the data-centric communication scheme and scarce energy and computation resources have a large impact on the design of routing protocols.

Data-centric communication leads to routing protocols relaying packets according to their data contents instead of their destination addresses, which alleviates the need of maintaining unique node identifiers throughout a network. Forwarding packets according to their contents requires a unified naming scheme so that intermediate nodes can access packet contents. The naming scheme, e.g. the attribute-value proposed by Directed Diffusion, offers additional benefits such as a powerful interface with applications. It also makes in-network processing possible, which reduces the number of exchanged messages through data aggregation.

Scarce energy and computation resources require that routing protocols keep complexity and power consumption to a minimum while relaying packets to maximize the lifetime of the network. Minimizing energy consumption can be realized by reducing the overhead of the protocol such as the number of exchanged routing messages or by selecting energy-efficient routes according to explicit energy-related metrics. Research results reported in

3 Routing Protocols

this chapter show that an energy-related metric should combine the energy drained on a link to minimize the cost per transmitted packet and the residual energy of nodes to delay network fragmentation.

Data-centric communication and energy efficient routing techniques are complementary. In Chapter 8, we present a routing protocol, called $O(1)$ -reception routing, that combines these techniques: it uses a combined energy-related metric and reduces the number of exchanged routing messages while following the concept of data-centric communication.

Part II

Contributions

4 Coping with Collisions

Collisions stem from simultaneous multiple access to a shared communication medium. In wireless networks, they occur when nodes use a contention-based medium access method. Specifically, a collision may happen when a receiver is within the transmission range of two or more nodes that are transmitting simultaneously so that it does not capture any frame. Each collision represents unnecessary energy dissipation. Therefore, reducing collisions should be one of the main design objectives of an access method for wireless sensor networks. Although there are schedule-based TDMA-like methods ([82, 53]) that are collision-free, contention-based methods ([41, 83, 39, 42, 25]) are still widely used in sensor networks, because they are less complex, they adapt well to traffic changes and network dynamics, and they do not require tight synchronization between nodes. Moreover, contention-based methods are more suitable for unlicensed radio bands.

In contention-based methods with carrier sensing before transmission, collisions may be caused by two types of nodes: visible nodes and hidden nodes [84]. A collision caused by a visible node occurs when two nodes perform carrier sensing at the same time, detect that the channel is free and transmit at the same time. A collision caused by a hidden node occurs when a node performs a carrier sense and does not detect the ongoing transmissions with which it may interfere, because their signal strength is below its carrier sense threshold. As the node does not detect these signals, it falsely assesses the channel as free and transmits, causing a collision.

In this chapter we study both types of collisions. In Section 4.1, we show that advocated solutions for coping with hidden node collisions are unsuitable for sensor networks. In Section 4.2, we model both types of collisions and derive closed-form formula giving the probability of hidden and visible node collisions. To reduce these collisions, we propose two solutions in Section 4.3. The first one based on tuning the carrier sense threshold saves a substantial amount of collisions by reducing the number of hidden nodes. The second one based on adjusting the contention window size is complementary to the first one. It reduces the probability of overlapping transmissions, which reduces both collisions due to hidden and visible nodes. In Section 4.4, we validate and evaluate the performance of these solutions through simulations.

4.1 Inefficiency of Existing Solutions for Sensor Networks

The problem of hidden node collisions has been extensively treated in the literature, however there is no sufficiently efficient for sensor networks. The main solution to the problem of hidden nodes when assuming a single channel is the RTS/CTS handshake proposed in MACA [85]. The RTS/CTS exchange reserves the channel both around the sender and around the receiver to protect a transmission from being corrupted by hidden nodes. Although the use of RTS/CTS lowers hidden node collisions in wireless networks, it is ineffec-

tive in multihop sensor networks for the following reasons.

RTS/CTS are control frames, therefore their transmissions is considered as an extra overhead. The RTC/CTS exchange may generate high overhead: about 40% to 75% of the channel capacity [40, 41]. Moreover, as RTS/CTS are broadcast, the energy drained by their transmissions may be considerable in preamble sampling protocols [41, 83, 25, 24, 11, 28, 26, 27].

Data frames in sensor networks are usually small; therefore, they have nearly the same size as RTS/CTS frames. In this case, the collision probability is nearly the same for data frames as for RTS/CTS. Thus, the probability that a communication is successful is higher when RTS/CTS are not used—when CTS/RTS are used, the communication is successful only if all RTS, CTS, and data frames are not corrupted, which is lower than the probability that the data frame alone is not corrupted.

RTS and CTS are broadcast frames. For some protocols, a unicast costs less energy than a broadcast [24, 25, 26, 27, 28, 11]. Thus, sending unicast data without RTS/CTS is much more beneficial.

RTS/CTS exchange does not avoid collisions in multi-hop networks [86].

RTS/CTS exchange may lower the network capacity due to the exposed node problem [87].

RTS/CTS exchange cannot be used for protecting broadcast frames.

As the use of RTS/CTS is unsuitable for multihop sensor networks, we to model the collisions and provide solutions, which are described in the next section.

4.2 Modeling Hidden and Visible Nodes Collision

4.2.1 System Model

We consider a sensor network in which node A wants to transmit a frame to node B (see Figure 4.1). We assume the following propagation model (see Table 4.1 for the notation):

$$P_{rx}(B) = \frac{P_{tx}(A)}{\alpha \cdot d(A, B)^\beta} \quad (4.1)$$

This expression described in (4.1) is generic. It covers two common channel models summarized in Table 4.2.

We define the following sets of nodes:

$N_{tx}(A)$: the set of nodes able to detect transmissions of node A :

$$N_{tx}(A) = \{x | d(x, A) \leq E\}, \quad (4.2)$$

where E is the *transmission range* defined as:

$$E = \sqrt[\beta]{\frac{P_{tx}(A)}{\alpha \cdot TRCS}}. \quad (4.3)$$

The nodes are inside the dotted circle in Figure 4.1.

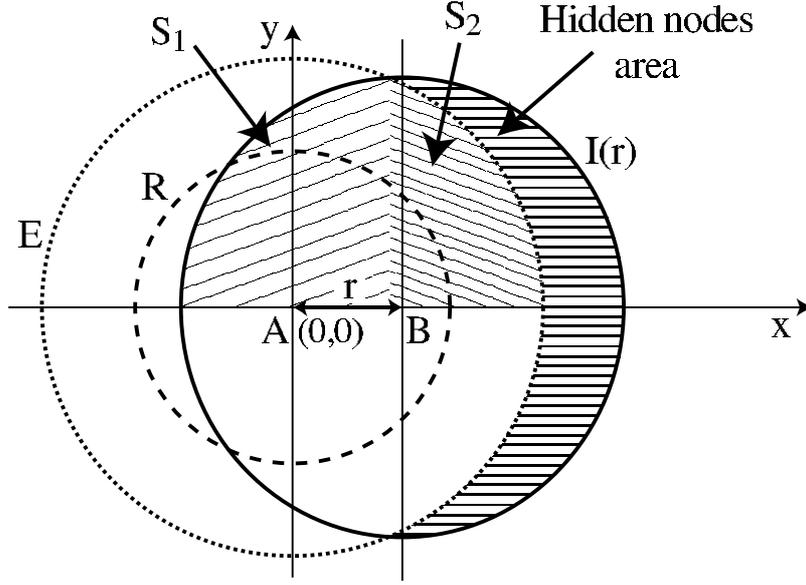


Figure 4.1: Transmission, reception, and interference ranges.

$N_{rx}(A)$: the set of nodes able to correctly receive frames sent by A in the absence of interference:

$$N_{rx}(A) = \{x | d(x, A) \leq R\}, \quad (4.4)$$

where R is the *reception range* defined as:

$$R = \sqrt[\beta]{\frac{P_{tx}(A)}{\alpha \cdot TR_{RX}}}. \quad (4.5)$$

A node outside this set cannot correctly decode the frames because of insufficient signal strength. This set is delimited by the dashed circle in Figure 4.1.

$N_i(A, B)$: the set of nodes that can corrupt a frame sent by A to B ($r = d(A, B)$):

$$N_i(A, B) = \{x | d(x, B) \leq I(r)\}, \quad (4.6)$$

where $I(r)$ is the *interference range*. The frame transmitted by A to B will be corrupted if:

$$\frac{P_{rx}(A)}{P_{rx}(x)} \leq TR_{CP} \quad \text{i.e.} \quad \frac{\frac{P_{tx}(A)}{\alpha \cdot r^\beta}}{\frac{P_{tx}(x)}{\alpha \cdot d(x, B)^\beta}} \leq TR_{CP} \quad (4.7)$$

By assuming that all nodes transmit with the same power, i.e. $P_{tx}(A) = P_{tx}(x)$, Eq. (4.7) rewrites as:

$$\left(\frac{d(x, B)}{r}\right)^\beta \leq TR_{CP} \quad (4.8)$$

Table 4.1: Notation for the analysis

$d(x, y)$	distance between nodes x and y (m)
r	distance between nodes A and B , $r = d(A, B)$ (m)
D	density of nodes, number of nodes per m^2
$P_{tx}(x)$	Transmission power of node x (Watt)
$P_{rx}(x)$	Received power at node x (Watt)
λ	Wavelength (m)
α	Channel gain, assumed constant ($m^{-\beta}$)
β	Path loss exponent
E	Signal detection range (m)
R	Signal reception range (m)
$I(r)$	Signal interference range (m)
TR_{CS}	Carrier sense threshold (Watt)
TR_{RX}	Reception threshold (Watt)
TR_{CP}	Threshold of capture ratio
G_t	The antenna gain at the transmitter
G_r	The antenna gain at the receiver
H_t	The antenna height at the transmitter (m)
H_r	The antenna height at the receiver (m)

Table 4.2: Channel models used for the analysis

Free Space	Two Ray Ground Reflection
$\alpha = \frac{(4\pi)^2}{\lambda^2 G_t G_r}$	$\alpha = \frac{1}{G_t G_r H_t^2 H_r^2}$
$\beta = 2$	$\beta = 4$

Therefore,

$$d(x, B) \leq r \sqrt[\beta]{TR_{CP}}, \quad (4.9)$$

Finally, the interference range can be obtained by combining Eq. (4.4) and Eq. (4.9):

$$I(r) = r \sqrt[\beta]{TR_{CP}}. \quad (4.10)$$

Note that the cardinality of this set depends on the distance between A and B .

$N_v(A, B)$: the set of nodes for which A is *visible*:

$$N_v(A, B) = N_{tx}(A) \cap N_i(A, B) \quad (4.11)$$

A visible node may corrupt a frame sent by Node A to Node B if both the visible node and Node A perform a carrier sensing and transmit at the same time.

$N_h(A, B)$: the set of nodes for which A is *hidden*:

$$N_h(A, B) = N_i(A, B) \setminus N_v(A, B) \quad (4.12)$$

A hidden node is not able to detect transmissions of Node A ; thus, it always assesses the channel to be free whilst Node A is transmitting. When a hidden node transmits, it corrupts the transmission of Node A to Node B .

4.2.2 Calculating the Number of Hidden and Visible Nodes

Let us denote by $n_h(r)$ the number of hidden nodes (resp. $n_v(r)$ the number of visible nodes). We assume that nodes are distributed over a surface with a homogeneous density D . Thus, $n_h(r)$ is proportional to the area of the zone in which hidden nodes may appear.

Let $S(r)$ be the common area of the zones corresponding to $N_{tx}(A)$ and $N_i(A, B)$. The circles of radius E and $I(r)$ intersect at two points: $(u, -\sqrt{E^2 - u^2})$ and $(u, \sqrt{E^2 - u^2})$, where $u = \frac{E^2 + r^2 - I(r)^2}{2r}$. We have,

$$S(r) = 2 \cdot [S_1(r) + S_2(r)], \quad (4.13)$$

where

$$S_1(r) = \int_{-I(r)+r}^u \sqrt{I(r)^2 - t^2} dt = I(r)^2 \left[\frac{\pi - a_1}{2} + \frac{\sin 2a_1}{4} \right],$$

and,

$$S_2(r) = \int_u^E \sqrt{E^2 - t^2} dt = E^2 \left[\frac{a_2}{2} - \frac{\sin 2a_2}{4} \right], \quad (4.14)$$

where $a_1 = \arccos \frac{u-r}{I(r)}$ and $a_2 = \arccos \frac{u}{E}$.

Therefore, we obtain the following results.

Proposition 1 *The number of hidden nodes is:*

$$n_h(r) = \begin{cases} 0 & \text{if } E \geq I(r) + r, \\ \pi \cdot [I(r)^2 - E^2] \cdot D & \text{if } E \leq I(r) - r, \\ [\pi \cdot I(r)^2 - S(r)] \cdot D & \text{otherwise} \end{cases} \quad (4.15)$$

Proposition 2 *The number of visible nodes is:*

$$n_v(r) = \pi \cdot I(r)^2 \cdot D - n_h(r). \quad (4.16)$$

4.2.3 Numerical Results for Different Radio Technologies: Bluetooth, ZigBee, WaveLAN

In this section, we calculate the number of hidden nodes with three popular radio technologies that are Bluetooth (IEEE 802.15.1), ZigBee (IEEE 802.15.4), and WaveLAN. Table 4.3 presents their parameters that come from the specifications of industrial products or IEEE standards¹.

¹For TR_{RX} , the IEEE 802.15.4 standard recommends the value of -85dBm, whereas the ZigBee compatible Freescale MC13192 transceiver uses -92dBm. We use the values encoded in `ns2` corresponding to the physical specifications of 914MHz Lucent WaveLAN DSSS. We theoretically calculate the carrier sense threshold TR_{CS} for the ZigBee and Bluetooth radios according to the relation described in Appendix.

Table 4.3: *Radio Parameters*

	Bluetooth (802.15.1)	ZigBee (802.15.4)	WaveLAN 914 MHz
P_{tx}	0 dBm	0 dBm	24.5 dBm
TR_{RX}	-80 dBm	-92 dBm	-64.4 dBm
TR_{CP}	11 dB	10 dB	10 dB
TR_{CS}	-102 dBm	-99 dBm	-78 dBm
$G_t(= G_r)$	1	1	1
$H_t(= H_r)$	0.1 m or 1.5 m	0.1 m or 1.5 m	0.1 m or 1.5 m

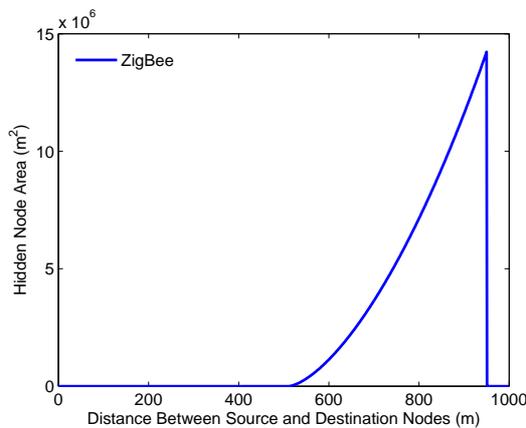
**Figure 4.2:** *Hidden nodes area with a Free Space model.*

Figure 4.2 shows the area that contains hidden nodes in function of the distance between the sender and the receiver for the Free Space model. Even though this model is purely theoretical, we can observe that with an antenna height of 0.1m, there are no hidden nodes for Bluetooth and WaveLAN. However, the hidden node area is important for ZigBee when the distance between the transmitter and the receiver is between 500m and 1000m.

Figures 4.3(a) and 4.3(b) show the hidden node area when assuming the Two Ray Ground Reflection model. We notice that there are hidden nodes only when WaveLAN or ZigBee radios are used. When a Bluetooth radio is used, there will be no hidden nodes as all the nodes will be visible.

4.3 Reducing Collisions

In this section, we propose two solutions for reducing collisions in wireless multihop networks. The first solution, based on tuning the carrier sense threshold, saves a substantial amount of collisions through reducing the number of hidden nodes. The second solution, based on adjusting the contention window size, is complementary to the first one. It lowers the probability that two transmissions overlap, which reduces the collisions caused by visible nodes. In addition, it also reduces the collisions caused by hidden nodes in case of flooding.

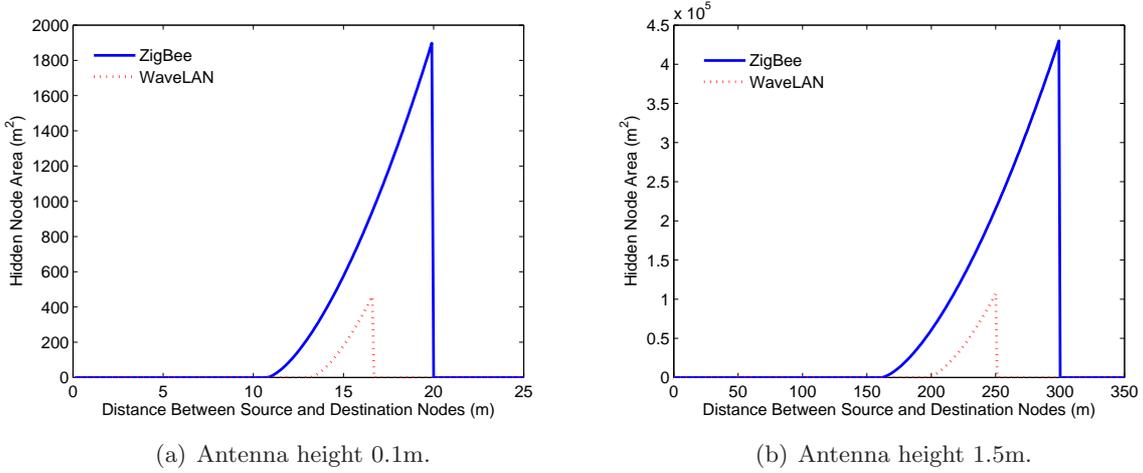


Figure 4.3: *Hidden nodes area with a 2-Ray Ground Reflection model.*

For example, assume that a node broadcasts a messages that is intended to be flooded. The neighbors that receive this messages will attempt to rebroadcast it. These neighbors may be hidden from one another. Therefore, a well adjusted contention window lowers the probability that transmission by these neighbors overlap, which reduces hidden node collisions.

In the following sections, we describe these two solutions in details.

4.3.1 Tuning Carrier Sense Threshold

Hidden nodes cause collisions because the transmission range, referred to as E , is not large enough for them to detect all the ongoing transmissions they may collide with. Therefore, to avoids such collisions, the transmission range must be sufficiently large so that the hidden node area (shown in Figure 4.1) becomes null.

There are two ways to increase the transmission range E : either increasing the transmission power P_{tx} or lowering the carrier sense threshold TR_{CS} (see Eq. 4.3). In the following, we assume that the transmission power is fixed, and we find the optimal carrier sense threshold that makes the number of hidden nodes equal to zero.

According to Proposition 1, the number of hidden nodes becomes null when $E \geq I(r) + r$, where r is the distance between the two nodes involved in the active transmission. From (4.3) and (4.10), we can write

$$\sqrt[\beta]{\frac{P_{tx}(A)}{\alpha TR_{CS}(r)}} \geq r \cdot \sqrt[\beta]{TR_{CP}} + r \quad (4.17)$$

Therefore, it is sufficient to take,

$$TR_{CS}(r) = \frac{P_{tx}(A)}{\alpha (r \cdot \sqrt[\beta]{TR_{CP}} + r)^\beta} \quad (4.18)$$

4 Coping with Collisions

To guarantee the non-existence of hidden nodes for all potential transmissions between any two nodes in the networks, r should be set to the maximum reception range R . Thus, the carrier sense threshold corresponds to $TR_{CS}(R)$.

Although tuning the carrier sense threshold reduces hidden node collisions, it has some shortcomings and limitations. First, it forces nodes to behave in a conservative way—many transmissions may be delayed, because a receiver will often detect a carrier due to its large radio carrier sense range, which increases the end-to-end delay and reduces the throughput of the network. Next, the threshold of carrier sensing cannot be reduced below a certain physical limit set by the noise of the receiver and the environment. Finally, even if the carrier sense threshold is set to its optimal value, hidden nodes may still exist: obstacles between nodes may change channel characteristics so that some signals are weakened and cannot be detected.

4.3.2 Adjusting Contention Window

This method aims at reducing collisions by choosing a larger contention window so that the probability of two transmissions overlapping is reduced. As increasing the contention window increases the delay and reduces the throughput, we should use the smallest contention window that keeps the probability of collision below a certain value. We can find a closed-form formula giving the smallest contention window according to a pre-set collision probability for access methods that use a similar contention mechanism as the 802.11 DCF (e.g. SMAC [39], TMAC [42], etc). This assumption allows us to use the Bianchi's results described in [88]. Thus, according to [88], the probability τ that a node transmits in a slot is

$$\tau = \frac{2}{CW + 1} \quad (4.19)$$

We use this result to compute p_c , the probability that a transmission attempt in a given slot ends up as a collision involving either a visible node or a hidden node.

We consider that each slot is composed of two phases: a node first performs a carrier sense for a duration of T_{CS} and then transmits in the same slot if the channel is free². In this case, only visible nodes that perform a carrier sense at the same instant may collide as they observe the channel free at the same time.

We call p_s the fraction of visible nodes that may cause a collision. Assuming that nodes have independently distributed time references and that they need to listen to the channel for at least the entire t_{CS} interval to detect an ongoing transmission, then $p_s = 2 \times \frac{T_{CS}}{T_{SLOT}}$. In this case, a transmission is successful if:

1. No node, among $n_v(r)$ nodes, transmits in the same slot. We call this probability P_V . We have,

$$P_V = (1 - \tau)^{n_v(r) \times p_s} \quad (4.20)$$

²This mechanism marginally extends the backoff between transmissions, but we neglect its impact on the transmission probability used below.

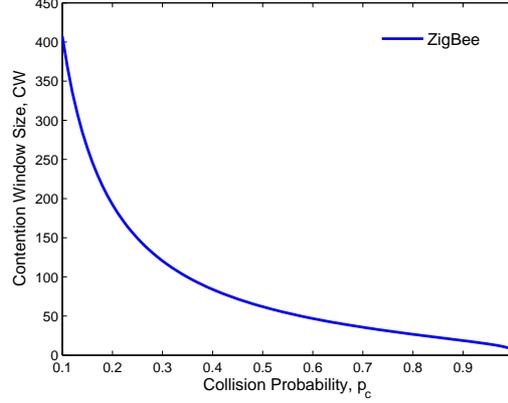


Figure 4.4: *Contention Window in function of collision probability for the ZigBee radio with antenna height of 0.1m.*

2. No node, among $n_h(r)$ nodes, transmits in the same slot. We call this probability P_H . We have,

$$P_H = (1 - \tau)^{n_h(r)} \quad (4.21)$$

By combining (4.20) and (4.21), we find p_c :

$$p_c = 1 - P_H P_V = 1 - (1 - \tau)^{n_h(r) + n_v(r) \times p_s}, \quad (4.22)$$

By substituting τ by its value given in (4.19), we obtain:

$$\left(\frac{CW - 1}{CW + 1} \right)^{n_h(r) + n_v(r) \times p_s} = 1 - p_c, \quad (4.23)$$

Finally, we get the expression that determines CW in function of r :

$$CW(r) = \frac{1 + \sqrt[n_h(r) + n_v(r) \times p_s]{1 - p_c}}{1 - \sqrt[n_h(r) + n_v(r) \times p_s]{1 - p_c}}. \quad (4.24)$$

Notice that the contention window CW depends on r , the distance between the sender and the receiver. Applying this result for controlling CW is quite difficult because all nodes in the network should know the distance between nodes willing to communicate. To avoid this problem, we can use a static value of CW by taking $r = R$, which corresponds to the worst case when the distance between nodes is equal to the signal reception range R . In this case, the contention window becomes:

$$CW(R) = \frac{1 + \sqrt[n_h(R) + n_v(R) \times p_s]{1 - p_c}}{1 - \sqrt[n_h(R) + n_v(R) \times p_s]{1 - p_c}}, \quad (4.25)$$

where $n = n_h(R) + n_v(R) \times p_s$. Figure 4.4 plots the result obtained in (4.25) according to the ZigBee radio parameters. It shows that the contention window should be exponentially increased to decrease p_c —the probability of collision.

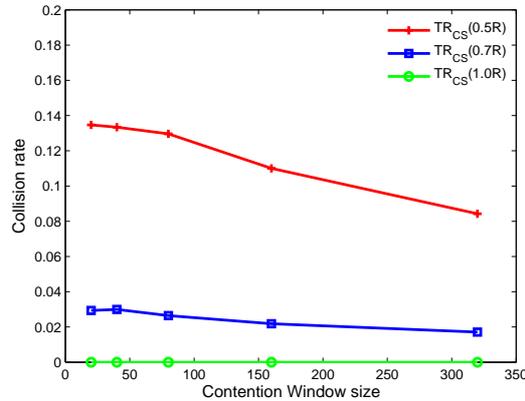


Figure 4.5: Collision probability due to hidden nodes.

4.4 Evaluation

In this section, we evaluate through ns2 [89] simulations the performance of the solutions described in the previous sections. Our goal for running simulations is to validate our results with an arbitrary network. We have set up the following simulation parameters:

we uniformly deploy 30 nodes in a $40\text{m} \times 40\text{m}$ square area,

we use the parameters of the Freescale’s MC13192 radio transceiver with a bandwidth of 250Kbps and a radio reception range R of about 20m (resulting from the Two Ray Ground propagation model with antenna height of 0.1m),

we randomly pick two nodes, a source and a destination, and make sure that they are not reachable in one hop,

the source node broadcasts 50 frames of 60 bytes at a constant bit rate (the inter-frame interval is set to 2ms),

each node re-broadcasts only once the frame it receives,

we use a 802.11-like MAC protocol with a slot time of $32\mu\text{s}$,

we set three different values for the carrier sense threshold: $TR_{CS}(0.5R)$, $TR_{CS}(0.7R)$, and $TR_{CS}(R)$

each point in the figures represents the average of 10 simulation runs.

Figure 4.5 shows the observed collision rates due to hidden nodes. As expected, these rates strongly depends on the carrier sense threshold—the case ($TR_{CS}(R)$) shows that tuning the carrier sense threshold does eliminate collisions that might be caused by hidden nodes. However, as previously stated, such an increase of the carrier sense range may be not possible or not effective due to obstacles. For example, a reasonable value of the carrier sense threshold may be ($TR_{CS}(0.7R)$), which corresponds to a collision rate less than

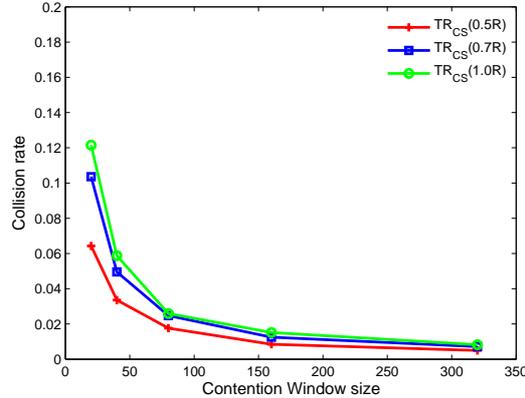


Figure 4.6: Collision probability due to contention.

5% (Figure 4.5). If smaller threshold values are to be used (e.g. $TR_{CS}(0.5R)$), then the contention window should be well chosen so that the collision rate is below a certain values (e.g. to get less than 10% of collisions, CW should be larger than 250 for slot times of 32μ , see Figure 4.5)

Figure 4.6 show an inverse phenomenon—the collision probability due to contention increases when the carrier sense threshold decreases. This means that even if tuning the carrier sense threshold has a beneficial effect on collisions caused by hidden nodes, it increases those caused by visible nodes. We can also see that when choosing a sufficiently large contention window, we can keep this type of collisions acceptably low.

4.5 Conclusion

In this chapter, we have analyzed the collision problem in multihop networks. We have classified the collisions into two types: those caused by visible nodes and those caused by hidden node. We have found closed-form formula for the probability of collisions in both cases. To reduce these collisions, we have proposed two solutions. The first one, based on tuning the carrier sense threshold, saves a substantial amount of collisions through reducing the number of hidden nodes. The second solution, based on adjusting the contention window size, is complementary to the first; it reduces the probability that two transmissions overlap, which reduces both collisions due to hidden and visible nodes.

4.6 Appendix

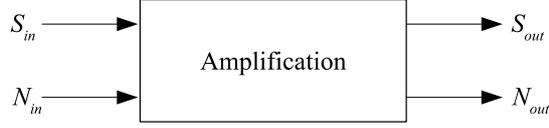


Figure 4.7: A reception system

The system described in Figure 4.7 has the following characteristics.

S_{in} is the strength of the received signal.

N_{in} is the strength of the noise of the system.

S_{out} is the strength of the signal after amplification.

N_{out} is the strength of the noise after amplification.

The Noise Factor F of this system is defined as:

$$F = \frac{SNR_{in}}{SNR_{out}} = \frac{S_{in}/N_{in}}{S_{out}/N_{out}} \quad (4.26)$$

The Noise Figure NF of this system is the Noise Factor converted to dB, i.e. $NF = 10 \log(F)$. We have,

$$NF = (SNR_{in})_{dB} - (SNR_{out})_{dB} \quad (4.27)$$

$$= (S_{in})_{dB} - (N_{in})_{dB} - (SNR_{out})_{dB} \quad (4.28)$$

Then,

$$(S_{in})_{dB} = NF + (N_{in})_{dB} + (SNR_{out})_{dB} \quad (4.29)$$

We have,

$$N_{in} = KTBw \quad (4.30)$$

where K is the constant of Boltzman, Bw is the bandwidth, and T is the temperature. At 25 °C, we have $KT = -174\text{dBm}$. If we have a channel of 1MHz ($Bw = 10^6$), then:

$$(N_{in})_{dB} = KT + 10 \log_{10}(10^6) \quad (4.31)$$

$$= -174 + 60 \quad (4.32)$$

$$= -114 \quad (4.33)$$

In systems currently in use, NF is approximately equal to 6dB.

In the computation of S_{in} , referred to as the TR_{CS} , we have taken $(SNR_{out})_{dB}$ equal to 6 dB. Therefore, we have the following values of TR_{CS} :

for Bluetooth, we have $TR_{CS} = -174 + 10 \log_{10}(10^6) + 6 + 6 = -102\text{dBm}$,

for ZigBee, we have $TR_{CS} = -174 + 10 \log_{10}(2 \cdot 10^6) + 6 + 6 = -99\text{dBm}$

4.6.1 Calculation of the Surface of Hidden and Visible Nodes

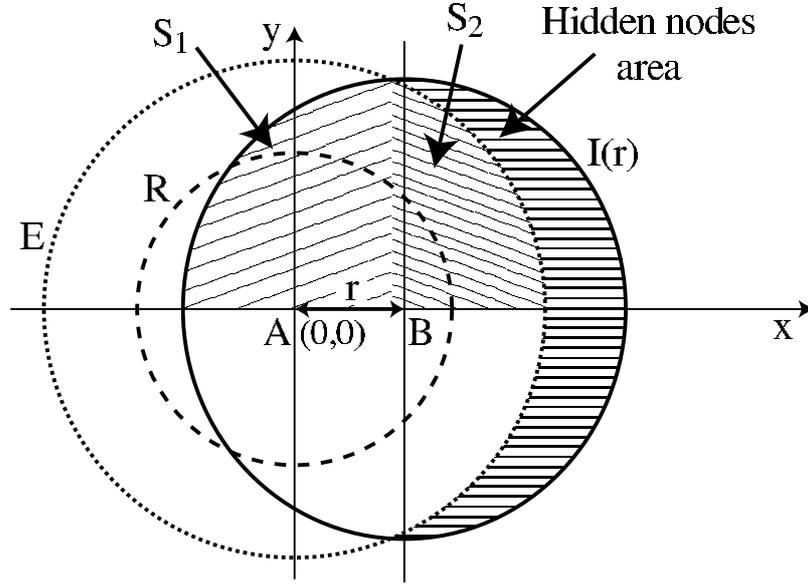


Figure 4.8: Transmission, reception, and interference ranges.

Let $H(r)$ be the zone of hidden nodes.

$$H(r) = \begin{cases} 0 & \text{if } E \geq I(r) + r, \text{ Interference(B)} \subset \text{Listening(A)} \\ \pi \cdot [I(r)^2 - E^2] & \text{if } E \leq I(r) - r, \text{ Interference(B)} \supset \text{Listening(A)} \\ [\pi \cdot I(r)^2 - S(r)] & \text{if } I(r) - r \leq E \leq I(r) + r, \text{ intersection calculate } u \end{cases} \quad (4.34)$$

In the third case, there is an intersection. Therefore, let's refer to $X(x, y)$ as the positive point of this intersection, i.e. $x > 0$ and $y > 0$.

We have the equations of the circles of transmission range and interference range as follows

$$x^2 + y^2 = E^2 \quad (4.35)$$

$$(x - r)^2 + y^2 = I^2(r) \quad (4.36)$$

Solving these equations gives:

$$x = \frac{E^2 + r^2 - I^2(r)}{2r} \quad (4.37)$$

$$y = \pm \sqrt{E^2 - x^2} \quad (4.38)$$

4 *Coping with Collisions*

5 Reducing Overhearing by Means of Abstract Frames

Overhearing is one of the main sources of energy dissipation in sensor networks. It occurs when a sensor node wastes energy while receiving useless frames or signals. There are many forms of overhearing, for example the reception of unicast frames destined to other nodes, or the reception of a long preamble (see Section 6). In this section, we focus on another form of overhearing, which is the reception of redundant broadcast frames.

The energy wasted in overhearing redundant frames may be important in sensor networks as broadcast in such networks is a frequent operation. Broadcast is typically used in many operations such as route discovery in on-demand routing protocols [59] or during the interest propagation phase in Directed Diffusion [61]. For all these operations, a given node in the network only needs to receive one broadcast message. All the subsequent broadcasts resulting from relaying by other nodes are redundant and thus useless, because they carry the same data contents. The reception of these redundant broadcast messages costs energy, therefore, it should be avoided.

To save the energy drained in receiving redundant broadcast messages, we propose *abstract frames*: an abstract frame is a small control frame sent before each broadcast frame. It contains a digest of the broadcast frame contents. A node listening to the channel uses the information in the abstract frame to identify and filter out redundant messages before their receptions. When a frame is expected to be redundant, the node switches its radio off to avoid its reception and thus to save energy. In Section 5.1, we present the key idea of abstract frames and discuss its contribution compared to other protocols that reduce the number of redundant transmissions. In Section 5.2, we analyze the performance of our abstract frames in terms of the lifetime extension compared with two MAC protocols: an *ideal* one that totally avoids idle listening, but does not filter out redundant messages at the MAC layer, and a *practical* one—SMAC [39]. In Section 5.3, we report simulation results on the performance of abstract frames method when used with SMAC.

5.1 Abstract Frames

5.1.1 Basic Idea

An abstract frame is a small frame sent immediately before each broadcast frame. It contains a digest of the contents of the subsequent data frame. A node uses the information in the abstract frame to learn about the subsequent data contents. If a node learns from the abstract frame that the data frame has already been received, it can switch its radio off, because the subsequent data is redundant as shown in Figure 5.1. In this way, a node only overhears redundant abstract frames instead of overhearing redundant data frames, which

5 Reducing Overhearing by Means of Abstract Frames

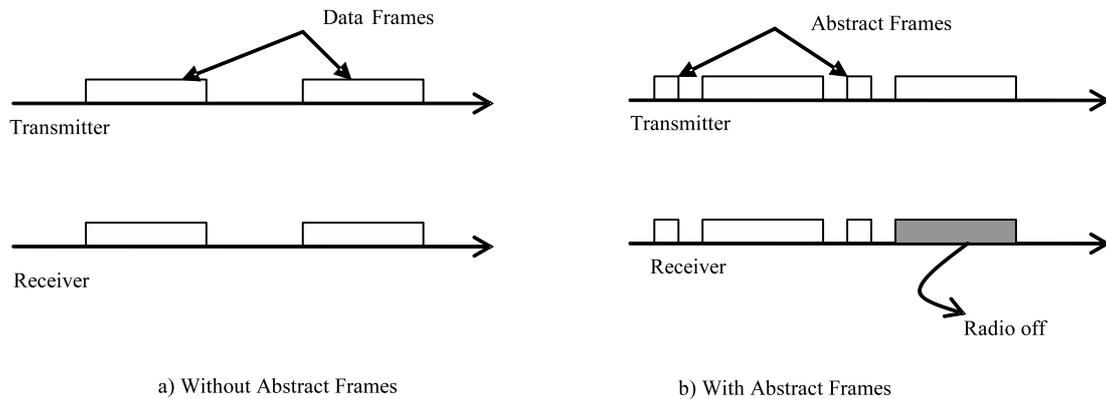


Figure 5.1: *Avoiding redundant frame reception by means of Abstract Frames.*

contributes to save more energy since abstract frames are expected to be far shorter than data frames.

An abstract frame has a *digest* field that contains either a unique identifier or a hash of the data contained in the subsequent broadcast frame. When the MAC protocol needs to transmit a frame, it constructs and transmits the corresponding abstract frame before transmitting the broadcast frame. Then, it inserts the digest field of the abstract frame in a table to avoid receiving it again from another node. This table logs frames that have been recently seen so that the MAC protocol may switch the radio off when it expects a redundant reception.

According to this procedure, the MAC layer always receives an abstract frame before a data frame for broadcast communications. It first checks in its table whether there is an entry with the same digest value. If such an entry exists, then the node switches its radio off to avoid receiving the same data again. However, if it does not exist, then the node continues to listen to the channel in order to receive the subsequent data frame. Once the node receives the data frame, it updates its table to avoid receiving redundant transmissions of the received data frame.

One can argue that the use of hash functions to calculate the digest values may lead to conflicts that cause a node to ignore a data frame that has not previously received — this happens when two different data have the same hash result. We think that such a situation is hardly likely to happen because of the following reasons. First, digest-field entries in the table of the MAC protocol are not permanent, but cleaned after a timeout value. Second, we can choose a suitable hash function and digest size so that collisions are very rare. A frame will be missed only if it involves two simultaneously active broadcasts with the same hash value during the timeout. Note that because broadcasts are not acknowledged, they are usually unreliable anyway.

For an efficient implementation of abstract frames, the CRC (Cyclic Redundancy Check) circuitry can be re-used to calculate the digest of data frames. The advantages of such a solution are twofold. First, it saves the overhead of calculating costly hash functions, which increases energy savings. Second, it does not require additional dedicated hardware for hash calculation, which does not increase the cost of the sensor node.

5.1.2 Discussion

There are several ways to reduce energy consumption caused by broadcasts. The most immediate one is to reduce the number of transmitted frames by avoiding redundant transmissions. Many proposed protocols select only a subset of nodes to flood a message while ensuring that all nodes eventually receive the message, e.g. CDS (Connected Dominating Sets) [90], MPR (Multi Point Relays) [91], or RNG (Relative Neighborhood Graphs) [92]. Other approaches optimize the transmission range by seeking a good trade-off between consuming more energy to reach more nodes with a small number of transmissions and reducing the transmission range and having more transmissions [17, 93, 94]. In any case, we can apply the abstract frame approach to all these protocols, because it reduces energy consumption at the MAC layer. However, as all of them try to reduce broadcast traffic, the more efficient they are, the less abstract frames are necessary.

5.2 Theoretical Performance

Although the use of abstract frames results in less energy consumption during the reception of redundant frames, it increases the energy drained per transmitted data frame. Therefore, we propose to analyze the performance of abstract frames taking into account these two parameters together. We propose to compare the lifetime of a node running a MAC protocol without abstract frames, which we call protocol P , to the lifetime of the same node when running protocol P with abstract frames, which we call P' .

We consider that all the communications are broadcast and nodes forward frames according to the flooding algorithm. As a candidate for protocol P , we take two examples. The first protocol is an ideal MAC protocol that does not have idle listening. The second protocol is SMAC that reduces idle listening via active/sleep schedules. For the sake of simplicity, we do not consider collisions in the following analysis.

We call E_P (resp. $E_{P'}$) the energy drained during a complete local flooding operation when nodes use protocol P (resp. P'). To quantify the ratio of lifetime extension by protocol P' compared to protocol P , we calculate the gain G_P defined in (5.1) for the two candidate MAC protocols.

$$G_P = \frac{E_P}{E_{P'}} \quad (5.1)$$

5.2.1 Abstract Frames with an Ideal MAC

To calculate the lifetime of a node, we consider a complete local flooding operation consisting in the reception of all frames from its neighbors and the forwarding of the broadcast frame exactly once. Thus, if the node has n neighbors, then the energy drained during the flooding operation is:

$$E_{ideal} = nT_{data}P_r + T_{data}P_t, \quad (5.2)$$

where T_{data} is the transmission time of the data frame and P_t (resp. P_r) is the power drained by a transmission (resp. a reception).

5 Reducing Overhearing by Means of Abstract Frames

When abstract frames are used, the node receives all abstract frames, but only one data frame; the node discards the other data frames because they are redundant. In addition, the node transmits one abstract frame before each data frame. The energy drained in this case is:

$$E_{ideal'} = (nT_{abstract} + T_{data})P_r + (T_{data} + T_{abstract})P_t, \quad (5.3)$$

where $T_{abstract}$ is the transmission time of an abstract frame. Finally, the lifetime extension is the following:

$$G_{ideal} = \frac{E_{ideal}}{E_{ideal'}} = \frac{(n + \rho)T_{data}}{(1 + \rho)T_{data} + (n + 1)T_{abstract}}, \quad (5.4)$$

where $\rho = P_t/P_r$

From Eq. (5.4), we conclude that the lifetime extension increases when the data size increases. It also increases when the number of neighbors increases. Note that the performance obtained with the use of abstract frames depends on the ratio $\frac{T_{abstract}}{T_{data}}$, which is the ratio of the abstract frame transmission time to the transmission time of data frames: the smaller $T_{abstract}$ compared to T_{data} , the larger lifetime extension we get. We can calculate $T_{abstract}^{max}$, the maximum value of $T_{abstract}$ beyond which there is no gain in using abstract frames: we need that $G_{ideal} > 1$. By using (5.4), we get the following result:

$$T_{abstract}^{max} = \left(\frac{n - 1}{n + 1} \right) T_{data} \quad (5.5)$$

5.2.2 Abstract Frames with SMAC

We follow the same methodology to evaluate the lifetime extension ratio for SMAC. Figure 5.2 shows an example of a node with three neighbors. In SMAC, nodes alternate active periods during which they can communicate and sleep periods during which they switch their radios off to save energy. The ratio of the period durations is controlled by the MAC duty-cycle parameter. The duration of the active period depends on a couple of parameters such that the data transmission time and the slot time used in the backoff procedure when nodes contend for the channel. Note that SMAC protocol carefully chooses the duration of the active period so that it is large enough to hold a data transmission including contention. However, there is no guarantee that a local flooding operation fits a single active period¹. More details on SMAC operation have been presented in Chapter 2.

Let us call T_{active} the duration of the active period. The local flooding operation may fit into one active period or more, depending on many parameters like the number of nodes n . Let us assume that a node needs k active periods, ($k > 0$) to carry out the local flooding operation. Therefore, we have the following relations (see Figure 5.2):

$$E_{smac} = nT_{data}P_r + T_{data}P_t + [kT_{active} - (n + 1)T_{data}]P_i \quad (5.6)$$

¹For the sake of simplicity, Figure 5.2 shows that the local flooding operation fits one active period

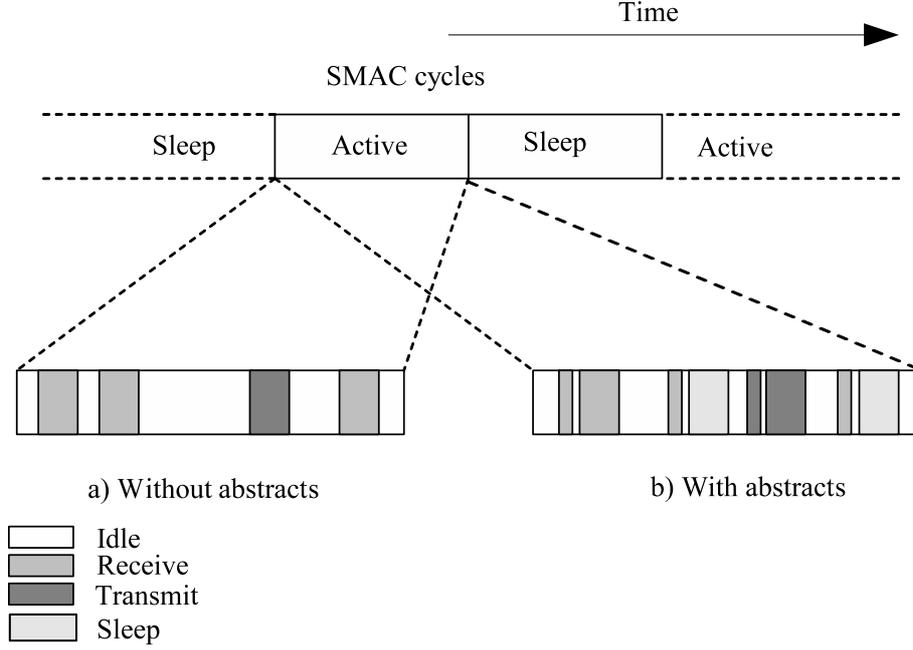


Figure 5.2: Operation of SMAC with and without Abstract Frames.

and,

$$E_{smac'} = (T_{data} + nT_{abstract})P_r + (T_{data} + T_{abstract})P_t + (n-1)T_{data}P_s + [kT_{active} - n(T_{data} + T_{abstract}) - (T_{data} + T_{abstract})]P_i, \quad (5.7)$$

where P_i (resp. P_s) is the power drained during the idle (resp. sleep) mode. In general, the power drained in the idle mode, in which the radio is ready to receive, is slightly less than the power drained during the receive mode. However, to simplify the comparisons, we will assume that $P_i = P_r$. In addition, the power drained during the sleep mode is negligible compared to other modes, thus we assume that $P_s = 0$. Therefore, we obtain the following lifetime extension for SMAC:

$$G_{smac} = \frac{E_{smac}}{E_{smac'}} = \frac{T_{data}\rho + kT_{active} - T_{data}}{(T_{data} + T_{abstract})\rho + kT_{active} - nT_{data} - T_{abstract}} \quad (5.8)$$

Eq. (5.8) shows that the gain depends on kT_{active} , the duration of active periods needed for the complete local flooding. These active periods include idle listening. To show the effect of idle listening on lifetime extension, we propose to rewrite Eq. (5.8) in function of T_{idle} , the amount of idle listening during the complete local flooding:

$$T_{idle} = kT_{active} - nT_{data} - T_{data} \quad (5.9)$$

Thus, Eq. (5.8) can be rewritten as:

$$G_{smac} = \frac{(n + \rho)T_{data} + T_{idle}}{(1 + \rho)T_{data} - T_{abstract}(1 - \rho) + T_{idle}} \quad (5.10)$$

5 Reducing Overhearing by Means of Abstract Frames

Eq. (5.10) shows that the gain decreases when idle listening increases. This is quite expected, because when idle periods dominate, the lifetime extension will not be significant.

Note that idle listening depends on traffic load: when the traffic load is high, nodes spend more time in transmit and receive modes, which decrease the amount of idle listening. Therefore, as Eq. (5.10) includes idle listening, it implicitly includes the effect of traffic load on lifetime extension. From Eq. (5.10), we conclude that the lifetime extension increases with the traffic load. However, an excessive traffic load causes collisions, which lowers the lifetime extension. We study this factor through simulation in Section 5.3.

Using the same approach as in Section 5.2.1, we calculate A_{max} for SMAC. We obtain,

$$T_{abstract}^{max} = \left(\frac{n-1}{\rho-1} \right) T_{data} \quad (5.11)$$

Interestingly, when $\rho \leq 1$, we have no constraint on the abstract frame transmission time to obtain a lifetime extension. Eq. (5.10) shows that the gain will always be larger than 1 when $\rho \leq 1$. In this case, the power drained in the transmit mode is less than the power drained in the idle mode. Thus, transmitting abstract frames saves more energy than remaining idle.

5.3 Simulation

The idea of avoiding redundant frames by means of abstract frames is generic and may apply to a large set of MAC protocols. We show in the next chapter how we use it with preamble sampling protocols. In this section, we focus on evaluating through ns-2 [89] simulations the use of abstract frames with MAC protocols based on using common active/sleep schedules. We have chosen SMAC to represent such a class of MAC protocols, because the code of SMAC under ns-2 is public and seems to be stable.

To quantify the lifetime extension achieved with the use of abstract frames, we propose to compare the lifetimes achieved by two MAC protocols: SMAC without abstract frames, and SMAC' which is SMAC with abstract frames. We carry out simulations to get more insight into the energy saving ratio since we have not taken all the parameters into account in the mathematical analysis, e.g. collisions.

Our application consists of simple flooding that forwards each new broadcast message it receives exactly once. The simulation scenario consists of one source node periodically broadcasting new messages. The other nodes flood the received broadcast message. The source node assigns a different message identifier for each new broadcast message. This identifier is used as our digest in the abstract frame. When the SMAC' agent receives an abstract frame, it checks whether it has already seen (sent or received) the data frame with the same message identifier. If the identifier is new, the SMAC' agent adds it to an internal table and keeps the radio on to receive the subsequent data frame. However, if the identifier has been already seen, then the following data is redundant and the SMAC' agent switches the radio off to save energy.

The application counts the number of non-redundant received messages. We use this number to quantify the lifetime of a node. The ratio of the number of received messages with SMAC' out of the number of received messages with SMAC determines the lifetime

extension. We have considered three situations to evaluate lifetime extension. These situations are the lifetime extension of the most vulnerable node, the lifetime extension of the most robust node, and the average lifetime extension of all the nodes. Results show that abstract frames extend the lifetime of SMAC by very close ratios in all three situations. Therefore, we have chosen to only analyze the results corresponding to the average lifetime extension of all the nodes.

We have measured the impact of data payload, traffic load, and network density on the lifetime extension. We have used a simple energy model to simulate a low power radio. In our energy model, the transmit mode uses 96mW, the receive mode uses 111mW and the ready-to-receive mode, that we also called idle mode, uses the same power as the receive mode. We have run our simulations with two different network topologies: a simple star topology to understand the results obtained and a randomly generated topology to measure the performance with realistic and more complex networks. The following sections report on the results obtained with these topologies.

5.3.1 Simple Star Topology

This topology consists of a source node placed in the center of the simulation area and some nodes placed around it. The number of nodes that are neighbors of the source node determines the density of the network.

For each simulation run, we calculate the lifetime extension as defined above. After some simulation runs, we calculate the average lifetime extension corresponding to a confidence level of 95%. We have found that the confidence interval remains fairly large even with a very large set of simulation runs, which reflects the fluctuations in lifetime extension ratios. These fluctuations mainly stem from the way used by SMAC to manage the active/sleep periods of border nodes². For example, in the implementation of SMAC we use, a border node remains active during all the active periods of the virtual clusters it belongs to. As a consequence, the actual MAC duty cycle of a border node increases, which decreases the efficiency of abstract frames as a longer duty cycle increases idle listening. Note that each time we run a simulation with a different random seed, we get different numbers of virtual clusters formed in the network. We have taken care to compare the lifetimes of SMAC and SMAC' in similar conditions, *i.e.*, for the same number of virtual clusters.

Figure 5.3(a) shows the histogram of the lifetime extension for various data frame sizes. The variability is due to the number of virtual clusters formed during each run. We plot this figure to point out the reason that causes the large confidence intervals presented in Figure 5.3(b).

In this first experiment, we set the traffic load to 0.1 messages per second. We notice that the lifetime extension is small for small data sizes, because the amount of time during which we switch the radio off to avoid redundant data receptions becomes negligible (around 4%)

²In SMAC, the active period is composed of two periods: synchronization period and data period. Nodes use the synchronization period to exchange SYNC frames from time to time in order to synchronize on a common sleep/wakeup schedule. Note that SMAC does not guarantee that all nodes synchronize on a single common schedule. Nodes that share a common schedule form a virtual cluster and the network may contain one or several virtual clusters. Some nodes, called border nodes, may belong to more than one virtual cluster. There are any options to manage the active/wakeup schedules of the border nodes; for more details, refer to Chapter 2.

5 Reducing Overhearing by Means of Abstract Frames

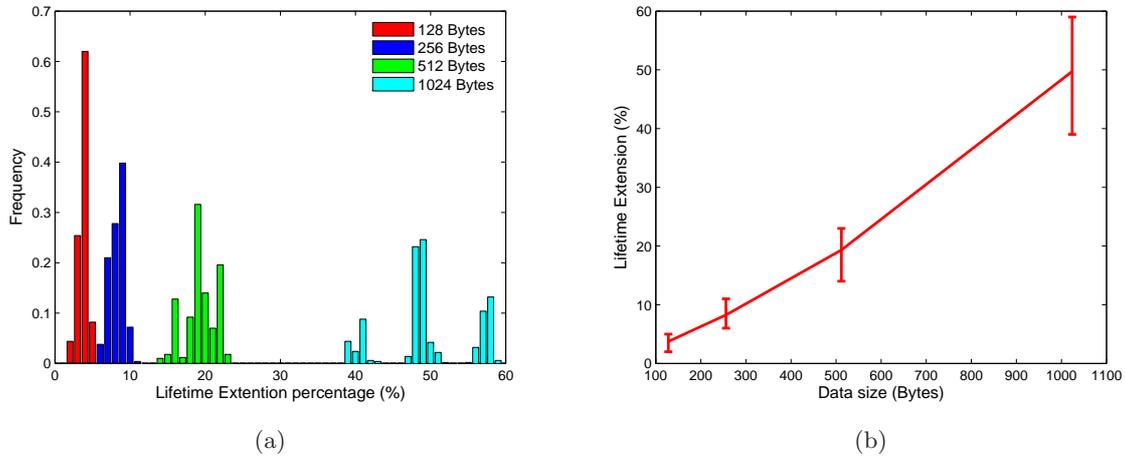


Figure 5.3: Lifetime extension according to various data frame sizes.

compared to the time the radio is on. However, when the data payload size increases to 1024 bytes, the lifetime extension increases by 40%-60%. This large confidence interval is due to variations of the number of virtual clusters in each simulation run. The ratio of 40% corresponds to the situation with many virtual clusters and the ratio of 60% corresponds to fewer virtual clusters. The formation of several virtual clusters decreases the lifetime extension, because idle listening becomes significant. We expect to obtain better performance with algorithms such as GSA (Global Schedule Algorithm) [34] that aim at making SMAC converge to one virtual schedule.

In Figure 5.4, we have varied the traffic load from 0.0125 to 0.1 messages per second *i.e.*, from one message every 10 seconds to one message every 80 seconds. The payload of the messages is 512 bytes. The figure shows that the lifetime extension increases when the traffic load increases. In this case, the time during which data are exchanged becomes non negligible compared to the duration of idle listening. Hence, the time during which SMAC' exploits abstract frames to switch off the radio becomes more significant. Note that we do not present results beyond 0.1 and below 0.0125 messages per second. We have observed a considerable increase of collision rates for traffic loads larger than 0.1 and negligible energy saving ratios for traffic loads less than 0.0125. We argue that this is rather SMAC-dependent and not a result showing intrinsic low performance of abstract frames. We expect to get better performance with other low power MAC protocols that manage idle listening in a better way like TMAC [42].

In Figure 5.5, we have varied the number of neighbors of the source node to simulate different network densities. This gives us a precise idea on what SMAC' is able to achieve in situations in which the channel is not saturated and collisions are rare. For these simple star topology networks, we have observed a collision ratio less than 1%, which allows us to see the effect of network density on lifetime extension independently from collisions. As we expected, the lifetime extension increases with network density.

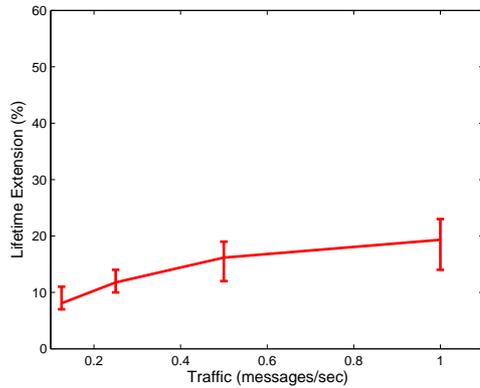


Figure 5.4: *Lifetime extension according to various traffic loads*

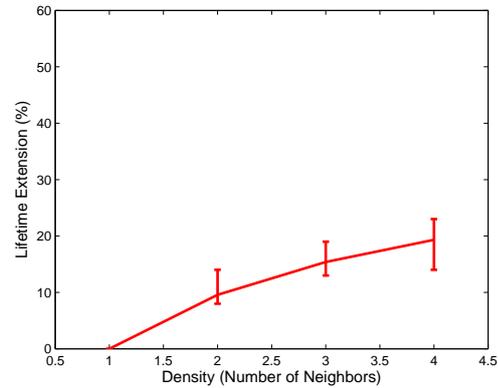


Figure 5.5: *Lifetime extension according to various number of neighbors*

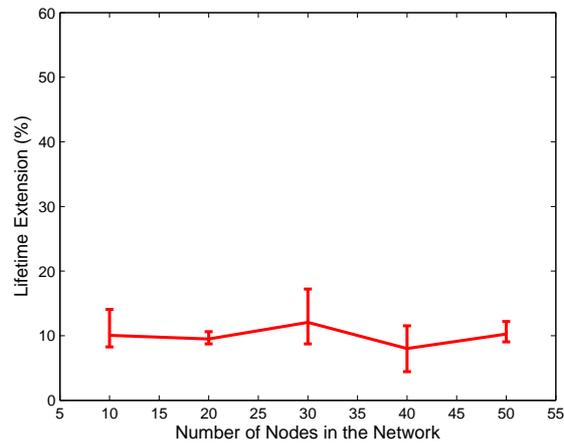


Figure 5.6: *Lifetime extension in function of the number of nodes in the network.*

5.3.2 Random Topology

In Figure 5.6, we have measured the lifetime extension ratios for more realistic topologies. We have generated five networks with node positions distributed uniformly in a square area except for the source node always placed in the center. The networks are connected and the degree of a network with less nodes is less than the degree of a network with more nodes. The average densities of the networks are: 1.8, 2.7, 3.3, 3.6 and 4.12 for networks with 10, 20, 30, 40, and 50 nodes respectively. In contrast to what one may expect, Figure 5.6 shows that the lifetime extension ratio does not systematically increase with the number of nodes, because higher densities of nodes increase the collision rates and the number of virtual clusters in the network, which reduces the performance of abstract frames method.

Collisions involve abstract frames in most of the cases and their rate increases when the density of nodes in the networks increases. When abstract frames collide, potential receivers will not know whether the subsequent data frames are redundant or not. Therefore, they

will not switch their radios off, which reduces the efficiency of abstract frames method. To get better performance, collisions should be reduced. A commonly used way to decrease collisions in this class of protocols is to increase the contention window size. This method may be not particularly efficient with SMAC, because nodes remain active during all the contention duration, which increases idle listening. Reducing collisions by increasing the contention window size is more efficient for protocols in which nodes sleep during the contention (e.g. MFP [83] described in Chapter 6, and the IEEE 802.15.4 [38]).

The number of virtual clusters in the network increases when the density of nodes increases. For the networks used in the simulations, having 10 (resp. 20, 30, 40 and 50) nodes, there are 1.3 (resp. 1.6, 1.8, 2.1 and 2.3) virtual clusters on the average. The duration of active periods of border nodes and thus idle listening increases when there are more virtual clusters in the network, which reduces the performance of abstract frames. As stated above, the reduced performance is SMAC-dependent and is not an inherent weakness of the abstract frames method. Improvements on SMAC such as GSA are expected to have better performance as they aim at reducing the number of virtual clusters.

5.4 Conclusion

We have presented a novel method to save energy by reducing the overhearing of redundant copies of broadcast frames. Our method consists in transmitting an abstract frame before each data frame: the abstract frame contains a digest of the data frame. Potential receivers use the abstract frame to learn about the subsequent data frame contents so that they can avoid its reception in case it is redundant.

We have evaluated the performance of our method analytically and by means of simulations in ns-2. Although we have applied abstract frames to SMAC, the key idea is generic and can be used in a large variety of MAC protocols including those based on preamble sampling. In the next chapter, we present MFP: an optimized preamble sampling MAC protocol that incorporates the idea of abstract frames to avoid the reception of redundant broadcast frames.

6 Micro-Frame Preamble MAC

MAC protocols based on preamble sampling techniques offer substantial energy savings for lightly loaded multihop sensor networks as they efficiently reduce idle listening. However, they suffer from a significant overhead due to irrelevant receptions. In preamble sampling protocols, two types of irrelevant receptions may occur: the reception to the whole preamble until the data arrives and the reception of useless data frames (e.g. a data frame destined to another node or a redundant copy of a previously received data frame).

To avoid these irrelevant receptions, we propose to add more information about the forthcoming data frame to the preamble transmitted before each data frame. This information allows a node to immediately learn the contents of the forthcoming data frame so that it decides whether to receive it or not without having to listen to the whole preamble. The MFP (Micro Frame Preamble) MAC protocol, presented in this chapter, is one implementation of this idea.

The rest of this chapter is organized as follows. In Section 6.1, we present the principles of MFP. In Section 6.2, we analytically evaluate the performance of MFP and compare it with traditional preamble sampling methods in terms of lifetime extension. We then evaluate the performance of MFP by simulation in Section 6.3. In Section 6.4, we show that MFP can be implemented with success on existing radio modules, such as Chipcon CC2500 and we show that the computation overhead of MFP is negligible. In Section 6.5, we MFP with other low power protocols. Finally, we conclude in Section 6.6.

6.1 Protocol Description

In preamble sampling protocols (see Chapter 2 for a more detailed description), as soon as a node detects a preamble, it keeps its receiver active until it gets the subsequent data frame. Without any particular optimization, this may consume large amounts of energy because on the average nodes receive half of the preamble each time they receive a data frame. Similarly, preamble sampling protocols have no means for identifying an irrelevant data-frame before its entire reception. These irrelevant data-frames may consume a significant amount energy, because their reception also includes the overhead of preambles as explained above.

To avoid these two types of irrelevant receptions, we propose *Micro-Frame Preamble* (MFP), in which we replace the traditional continuous preamble by a series of small frames called *micro-frames*. Each microframe contains an indicator of the data frame contents such as the destination address or a digest of the data field. In addition, each microframe contains a *sequence number* that indicates the number of microframes to be transmitted before a data frame. When a node receives a microframe, it can deduce from the sequence number when the forthcoming data frame will arrive. It also learns from the data-frame contents indicator whether the data frame is worth receiving: for example, a unicast frame destined to the node of interest or a broadcast frame not received yet. The main advantages

6 Micro-Frame Preamble MAC

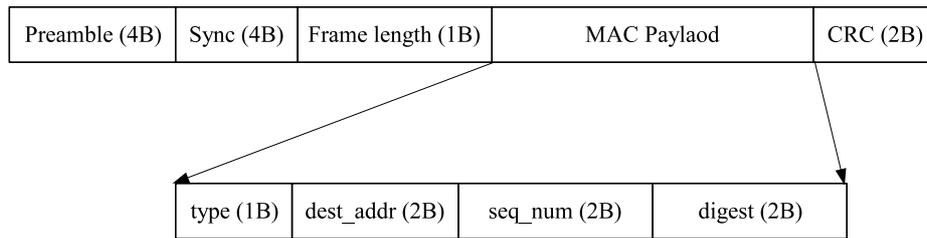


Figure 6.1: Microframe structure as implemented in the CC2500 module. Field sizes, in bytes, are in parentheses.

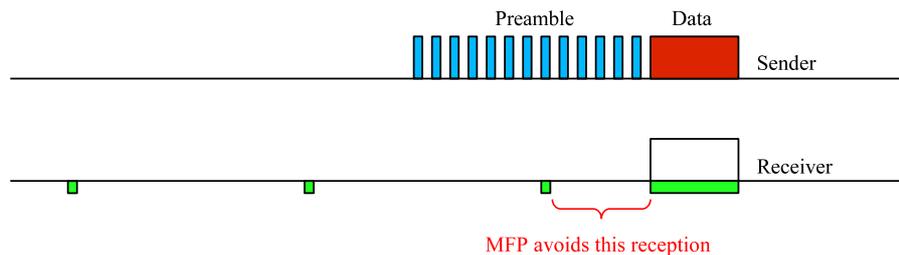


Figure 6.2: Avoiding reception of irrelevant preamble.

of MFP are discussed in the following sections.

6.1.1 Avoiding the Reception of the Whole Preamble

Figure 6.1 shows a proposed structure for microframes¹. A microframe contains a type field so that a node is able to distinguish microframes from other frames such as data or acknowledgments. It also contains a sequence number that indicates the number of microframes to be transmitted in the preamble before the data frame. The transmission of all microframes in MFP should last longer than the check interval CI . When a node wakes up and receives a microframe, it can infer from the microframe sequence number when the data frame will be transmitted relatively to the received microframe. This information enables a node to switch its radio off during the transmission of the subsequent microframes and to switch it on again just in time to receive the data frame. Figure 6.2 illustrates this operation.

6.1.2 Avoiding the Reception of Irrelevant Data

We assume a sensor network with simple applications that do not take advantage of over-hearing; that is, unicast frames that are destined to another node and redundant broadcast frames that have been previously received are irrelevant and thus should not be received.

A commonly used method for avoiding the reception of irrelevant unicast frames is the use of the RTS/CTS headers to learn about the destination of the subsequent data frame. Thereby, a node can go to sleep to save energy, if it is not the destination of the data

¹We present the details of our protocol as implemented on Chipcon CC2500 described later. Other implementations that may suit different MAC protocols are also possible.

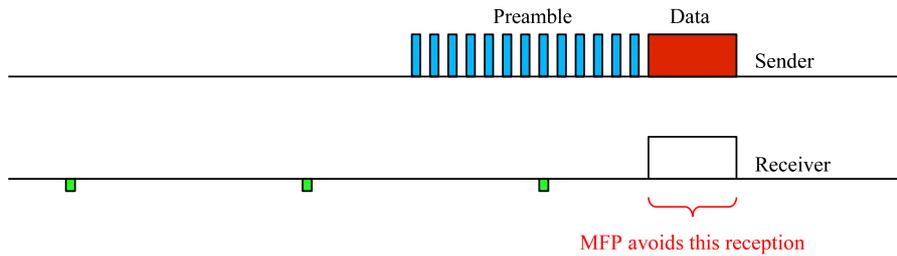


Figure 6.3: *Avoiding reception of an irrelevant data frame.*

frame. This idea, similar to what is done by the virtual carrier sense with NAV (Network Allocation Vector) in the IEEE 802.11 DCF [43], is unsuitable for preamble sampling protocols, because RTS/CTS are broadcast frames, thus they require full-length preambles for their transmissions, which results in high energy consumption both at the receiver and the transmitter.

Instead, we propose to put the destination address of the data frame into microframes. In this way, a node that wakes up to sample the channel receives a microframe from which it knows whether it is the destination of the subsequent data frame or not. Figure 6.3 shows an example of a node learning from a microframe that it is not the destination of the forthcoming data frame.

If the data frame is broadcast, the destination address used in microframes does not help the node to know whether the data frame is redundant or not. That is why we include a special field, called digest, that contains a digest of the data and makes it possible for a node to know about the contents of the data frame before entirely receiving it. To use this information, nodes should maintain a table keeping track of digests of previously seen (sent or received) frames. The use of such digest has been presented and evaluated in Chapter 5.

6.1.3 Other benefits

MFP also makes it possible to save energy when a node receives a microframe at the instant it performs a carrier sense before sending a frame. The node uses the sequence number and the destination address of the received microframe to estimate the schedule of data transmission including the acknowledgment. In this way, the node postpones its transmission until the end of the ongoing transmission, which saves the node the energy drained in potential repeated attempts for accessing the channel during the ongoing transmission.

6.2 Theoretical Evaluation

The main advantage of MFP compared to traditional preamble sampling schemes is the reduction of power consumption during reception. This is mainly due to minimizing the time during which the radio is on (in Receive mode) during preamble reception. In this section, we analytically compare the performance of MFP with other preamble sampling protocols. We take LPL (Low Power Listening) [41] as a representative of preamble sampling

6 Micro-Frame Preamble MAC

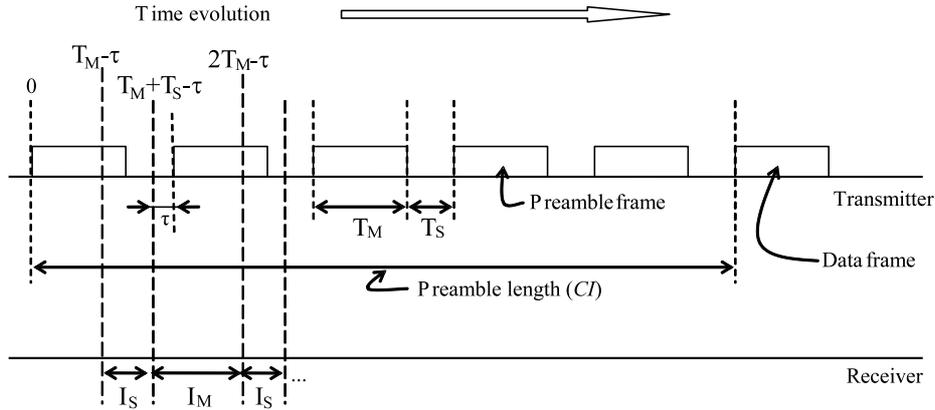


Figure 6.4: Transmission of microframes in the preamble with a gap between two consecutive microframes.

protocols². We start by modeling the reception duration of a node when it uses MFP or LPL. Then, we use this result to calculate the average lifetime duration of a node in both cases: MFP and LPL. Note that for the rest of the analysis, we do not consider the power drained in Sleep mode as it is generally negligible compared to that of other modes (4 orders of magnitude lower, see Table 6.2 for measured current consumption of CC2500).

6.2.1 Reception Cost

LPL

Let $T_{LPL}^r(t)$ be a random variable that corresponds to the time a node spends in Receive mode to receive one data frame. As each node chooses its schedule locally, a node may wake up at any random instant in $[0, T_{CI}]$, where T_{CI} is the duration of the Check Interval. In LPL, when a node wakes up and detects a preamble being transmitted on the channel, it keeps listening to the preamble until it receives a data frame. Therefore, we have:

$$T_{LPL}^r(t) = \tau + U_{[0, T_{CI}]}(t) + T_{data}. \quad (6.1)$$

where τ is the time needed to go from Sleep mode to Receive or to Transmit modes³, T_{data} is the time needed to receive one data frame, and $U_{[0, u]}(t)$ stands for a uniform random variable in $[0, u[$. Thus, $E[T_{LPL}^r(t)]$, the mean reception time for LPL is the following:

$$E[T_{LPL}^r(t)] = \tau + \frac{T_{CI}}{2} + T_{data}. \quad (6.2)$$

²Note that our model also applies to other preamble sampling protocols that are improvements on LPL. We discuss in the Section 6.5 how MFP improves these protocols.

³Actually, the transition from Sleep mode to Transmit or Receive modes consists of switching first from Sleep to Idle mode and then from Idle to Receive or Transmit modes. We have neglected the energy drained during the transition from Sleep to Idle mode. Note that τ is not negligible and smaller values for τ substantially increase the performance of preamble sampling protocols.

Therefore, the amount of energy drained during the reception with LPL is:

$$\mathcal{E}_{LPL}^r = E[T_{LPL}^r(t)]P_r. \quad (6.3)$$

where P_r is the power drained when the radio is in Receive mode.

MFP

Let $T_{MFP}^r(t)$ be a random variable that corresponds to the time spent in Receive mode to receive one data frame. As each node chooses its schedule locally, the node may wake up at any time during the transmission of the preamble: during I_S or I_M intervals as shown in Figure 6.4. We denote the duration of I_S by T_S (the inter microframe gap) and the duration of I_M by T_M (the microframe transmission time).

As shown in Figure 6.4, if the node wakes up during I_M , it misses the beginning of a microframe and it cannot decode it correctly⁴. In this case, the node suspects it has just lost a microframe, so it keeps listening to the channel until it receives the subsequent microframe. If the node finds the channel idle for a duration longer than T_S , then it goes back to sleep concluding that no frames are being transmitted. However, if the node wakes up during I_S , then it receives the subsequent microframe.

Therefore, if the node wakes up during I_S (resp. I_M) interval, then the time it spends in Receive mode to correctly decode a microframe is $\mu_S(t)$ (resp. $\mu_M(t)$):

$$\begin{cases} \mu_M(t) = \tau + U_{[0, T_M]}(t) + T_S + T_M. \\ \mu_S(t) = \tau + U_{[0, T_S]}(t) + T_M. \end{cases} \quad (6.4)$$

The probabilities q_S and q_M that a node wakes up during I_S and I_M intervals are the following, respectively:

$$\begin{cases} q_S = \frac{T_S}{T_S + T_M}. \\ q_M = \frac{T_M}{T_S + T_M}. \end{cases} \quad (6.5)$$

To evaluate the influence of receiving irrelevant data frames, we assume that the proportion of relevant data frames is α . It may depend on many parameters, for example the application traffic pattern. In the case of nodes running only a flooding application, the reception of only one message is sufficient, so all other copies of the same message subsequently forwarded by neighbors become irrelevant. Therefore, if a node has n neighbors, then all the $(n - 1)$ messages forwarded afterward are redundant. In this case, $\alpha = 1/n$.

When using the information about the data-frame contents in a microframe, a node that receives a microframe may sleep during the subsequent data frame transmission. As a node wakes up to receive the subsequent data frame only in α percent of cases, the time needed for a single message reception is:

$$T_{MFP}^r(t) = q_M \mu_M(t) + q_S \mu_S(t) + (\tau + T_{data})\alpha. \quad (6.6)$$

⁴We assume that a node will fail to decode a microframe if it misses the reception of its first bit. This assumption can be easily relaxed — we can easily assume that the node will fail to decode a frame correctly if it misses the x first bits of the preamble. In both cases, the analysis remains the same, only the durations of the intervals I_M and I_S change.

6 Micro-Frame Preamble MAC

where $q_M\mu_M(t) + q_S\mu_S(t)$ corresponds to the time needed for receiving one microframe and $(\tau + T_{data})$ to the time needed to receive one data frame (it includes τ , because we assume that the node is in Sleep mode after receiving a microframe). Thus, the average reception duration is

$$\begin{aligned} E[T_{MFP}^r(t)] &= q_M E[\mu_M(t)] + q_S E[\mu_S(t)] + (\tau + T_{data})\alpha \\ &= \tau + \frac{T_S}{2} + \frac{3T_M}{2} + (\tau + T_{data})\alpha. \end{aligned} \quad (6.7)$$

Therefore, the amount of energy drained in reception with MFP is

$$\mathcal{E}_{MFP}^r = E[T_{MFP}^r(t)]P_r. \quad (6.8)$$

6.2.2 Channel Sampling Cost

The average time a node spends in channel sampling and in transmission with LPL is the following:

$$T_{LPL}^s = \tau + T_{CS}. \quad (6.9)$$

where T_{CS} is the time needed to determine whether there is a preamble being transmitted on the channel.

In MFP, the channel sampling duration is slightly different, because of the possible inter microframe gap (I_S interval in Figure 6.4). Indeed, a node must sense the channel for a duration of T_S to determine whether the channel is free. Therefore, the duration of channel sampling in MFP is the following:

$$T_{MFP}^s = \tau + T_S + T_{CS}. \quad (6.10)$$

The amount of energy drained in channel sampling with LPL and MFP is:

$$\mathcal{E}_{LPL}^s = T_{LPL}^s P_{\text{samp}}. \quad (6.11)$$

$$\mathcal{E}_{MFP}^s = T_{MFP}^s P_{\text{samp}}. \quad (6.12)$$

where P_{samp} is the power drained when the radio is in channel sampling mode. This power is almost equal to the power in Receive mode. For the remainder of the analysis, we assume $P_{\text{samp}} = P_r$.

6.2.3 Transmission Cost

The energy drained in the transmission of one message in LPL is the following:

$$\mathcal{E}_{LPL}^t = (\tau + T_{CS})P_r + (\tau' + T_{CI} + T_{data})P_t. \quad (6.13)$$

where P_t is the power drained when the radio is in Transmit mode and τ' is the transition time from Receive mode to Transmit mode.

In MFP, the energy drained in the transmission of one message depends on N_{MF} , the number of microframes transmitted in the preamble. As the radio potentially goes to Sleep

mode between microframe transmissions, the number of microframes transmitted in the preamble is:

$$N_{MF} = \left\lceil \frac{T_{CI}}{T_S + T_M} \right\rceil. \quad (6.14)$$

Therefore, the energy drained during a transmission in MFP depends on the inter microframe time. If the inter microframe time is larger than the transition time τ , then the transmitter goes to Sleep mode between the transmission of two consecutive microframes. Otherwise, the transmitter does not go to Sleep mode and we assume that the energy drained during the inter microframe gap is equal to that drained in Transmit mode. Therefore, we have:

if $T_s < \tau$, then:

$$\mathcal{E}_{MFP}^t = (\tau + T_{CS})P_r + \left(\tau' + \left\lceil \frac{T_{CI}}{T_S + T_M} \right\rceil T_M + T_{data} \right) P_t. \quad (6.15)$$

if $T_s \geq \tau$, then:

$$\mathcal{E}_{MFP}^t = (\tau + T_{CS})P_r + \left(\tau' + \left\lceil \frac{T_{CI}}{T_S + T_M} \right\rceil (\tau + T_M) + T_{data} \right) P_t. \quad (6.16)$$

6.2.4 Modeling Node Lifetime

We define \mathcal{L}_\cdot , the lifetime duration of a node as

$$\mathcal{L}_\cdot = \frac{E_{\text{initial}}}{\mathcal{P}_\cdot}. \quad (6.17)$$

where \mathcal{P}_\cdot (joules/sec) is the average power consumed by the sensor node and E_{initial} (joules) is its initial energy. The dot ' \cdot ' can be either MFP or LPL. For the sake of conciseness and simplicity, we only consider the power consumed by the radio — the overhead of the microcontroller is very small as shown in Section 6.4.2. Therefore, we have

$$\mathcal{P}_\cdot = \mathcal{P}_\cdot^t + \mathcal{P}_\cdot^r + \mathcal{P}_\cdot^s. \quad (6.18)$$

where \mathcal{P}_\cdot^t (resp. \mathcal{P}_\cdot^r , \mathcal{P}_\cdot^s) is the average power drained in transmission (resp. reception, sampling).

The average power drained during preamble sampling is

$$\mathcal{P}_\cdot^s = \frac{\mathcal{E}_\cdot^s}{T_{CI}}. \quad (6.19)$$

Similarly, we compute the average power drained during transmission \mathcal{P}_t

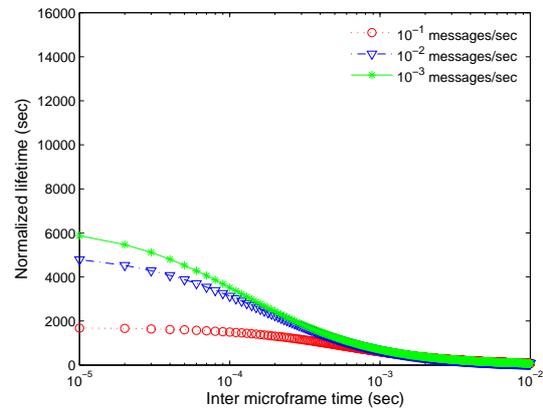
$$\mathcal{P}_\cdot^t = \frac{\mathcal{E}_\cdot^t}{T_{\text{traffic}}}. \quad (6.20)$$

and the average power drained during reception \mathcal{P}_r

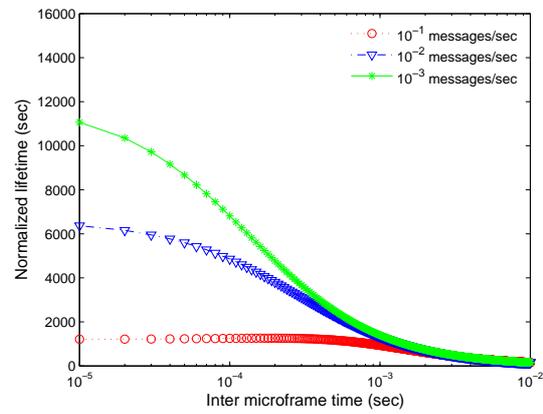
$$\mathcal{P}_\cdot^r = \frac{\mathcal{E}_\cdot^r \cdot n}{T_{\text{traffic}}}. \quad (6.21)$$

where T_{traffic} is the average inter data-frame transmission time that characterizes the application traffic load. For the derivation of the average power drained during reception (Eq. 6.21), we have assumed that nodes run a simple flooding protocol in which each node forwards a received message exactly once.

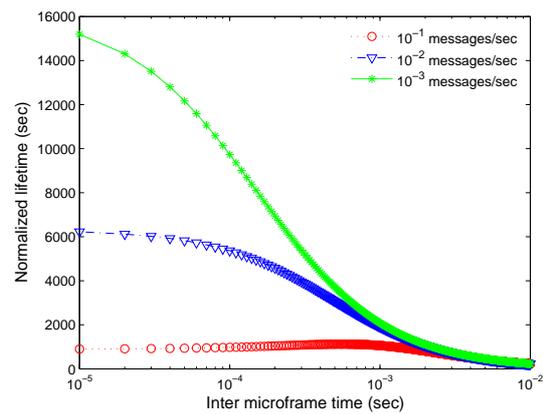
6 Micro-Frame Preamble MAC



(a) Check interval = 50ms



(b) Check interval = 100ms



(c) Check interval = 150ms

Figure 6.5: Normalized lifetimes of a node using MFP in function of different inter microframes gaps and various traffic rates.

6.2.5 Comparisons

To evaluate the performance of MFP and compare it with LPL, we plot Figure 6.5 and Figure 6.6 with the following parameters: the transition times τ and τ' are equal to $88.4\mu\text{s}$ and $9.6\mu\text{s}$ respectively, the transmission times of a 18 byte microframe and a 256 data frame at 250kb/s are equal to $T_M = 576\mu\text{s}$ and $T_{data} = 8.48\text{ms}$, respectively, and the carrier sense time T_{CS} is equal to $32\mu\text{s}$. For both figures, we plot a normalized lifetime corresponding to the lifetime of a node with an initial energy of $E_{initial} = 1$ Joule.

Figure 6.5 shows that a node that uses MFP benefits from longer lifetimes when the traffic load is low and the check interval is large. This is expected, because lower traffic loads imply less transmissions and larger check intervals imply longer sleep periods. Figure 6.5 also shows that shorter inter microframe gaps improve lifetimes, because the sampling time is smaller (T_{MFP}^s decreases when T_S decreases). The energy saved with smaller inter microframe gaps is larger when the traffic load is lower, because nodes spend more time in sampling than in transmitting messages.

The amount of energy saved in preamble sampling protocols depends on the duty cycle. To save more energy, nodes need to use lower duty cycles to spend more time in low-power mode. However, although lowering the duty cycle (i.e. increasing the check interval) does reduce idle listening (see Eq. (6.19)), it has some side effects: it increases the cost of transmissions (see Eq. (6.13)) and receptions (see Eq. (6.2)). Therefore, there is an optimal value for the check interval that achieves the best trade-off between the cost of transmission, reception, and channel sampling in function of the traffic generated by the applications.

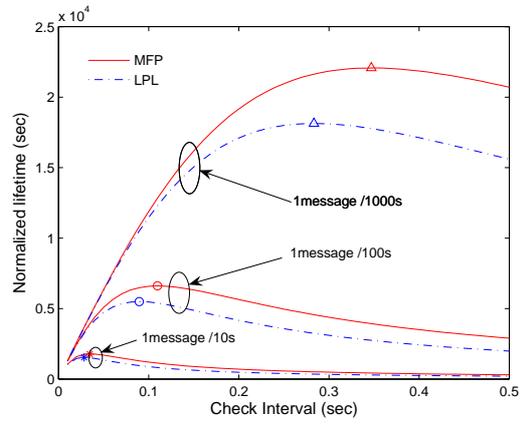
To obtain optimal lifetimes, we should use the smallest possible inter microframe gap (see Figure 6.5). In Figure 6.6, we show normalized lifetimes for MFP and LPL in function of the check interval. We can see that a node has a longer lifetime for MFP than for LPL for all the check intervals used. The figure also shows that optimal check intervals are larger for MFP, which means that MFP allows smaller duty cycles and then longer lifetimes. We can also see that the lifetime of a node that uses LPL decreases when the number of its neighbors increases, because the node spends energy in receiving redundant messages transmitted by its neighbors (each messages contains preamble plus data). However, when a node uses MFP, neither it does receive the preamble of these redundant messages nor does it receive the data. Therefore, its lifetime does not decrease. We can also notice that even if there are no irrelevant reception ($n = 1$), MFP presents a substantial gain of lifetime.

6.3 Simulations

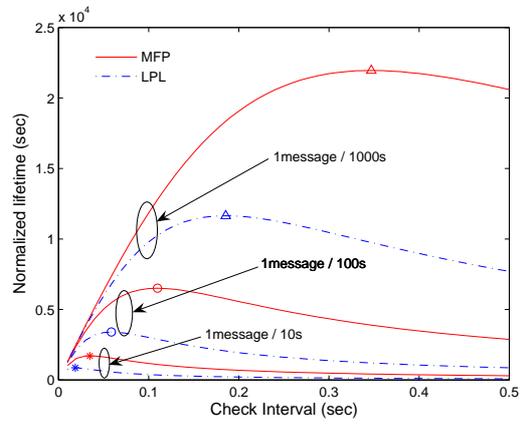
The analysis in Section 6.2 does not take collisions into account, because their modeling in wireless multihop networks is a difficult problem [95]. Therefore, actual lifetimes may differ from the values obtained analytically. To validate our analytical results, we have run simulations: we have implemented MFP and LPL in ns-2 [89] and compared their performance in large networks with random topologies. We have generated different topologies in which we randomly choose x and y coordinates of each node according to a uniform distribution. We make sure that each of the generated networks is connected, i.e. there is a path between any two nodes. Figure 6.7 shows an example of such a network.

To get into the details of the protocol behavior, we have run our simulation on a chosen

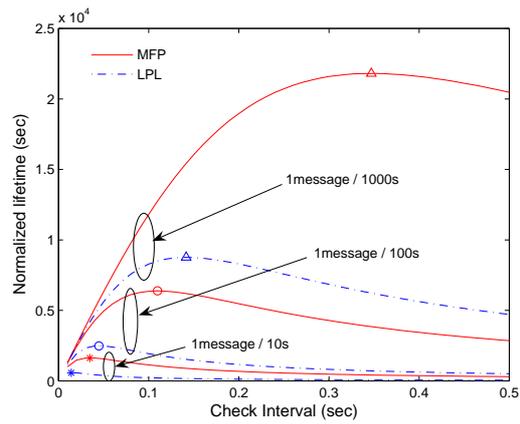
6 Micro-Frame Preamble MAC



(a) $n = 1$



(b) $n = 5$



(c) $n = 10$

Figure 6.6: Normalized lifetime of nodes for different check intervals. Microframes are transmitted without inter microframe gaps, i.e. $T_S = 0$.

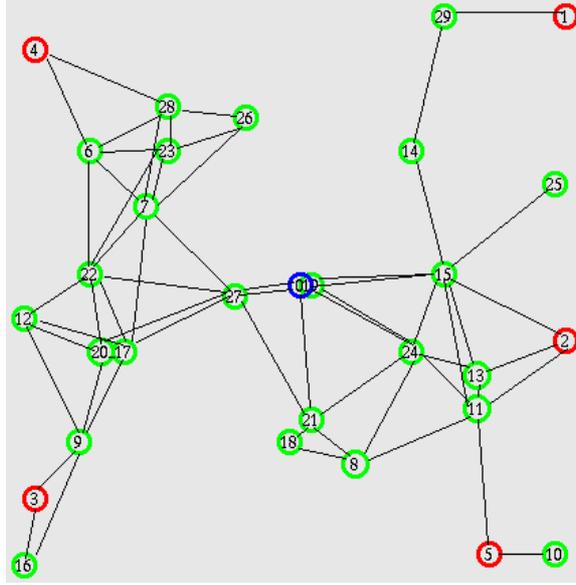


Figure 6.7: Chosen random topology: Node 0 is the sink, Nodes 1 through 5 are the sources, the other nodes (6 through 29) are intermediate nodes that relay traffic.

topology presented in Figure 6.7. It includes nodes with a very low degree of connectivity (e.g. nodes 1, 10, and 25 have only one neighbor) as well as a high degree of connectivity (e.g. node 15 has eight neighbors). In addition, there are several routes from the sources (nodes 1 through 5) to the sink (node 0).

We use the parameters of PHY and MAC layers for the CC2500 evaluation board [11]. We set the size of a microframe to 18 bytes that corresponds to $576\mu\text{s}$ at 250Kb/s. As ns-2 is a packet level simulator, a node that misses the first bit of a frame (e.g. because it is sleeping) will also miss all the subsequent bits of the same frame. That is why we require that carrier sense duration be longer than one microframe transmission time to guarantee that a node missing the first bit of a microframe can detect the first bit of a potential subsequent microframe (or a data-frame).

We consider three MAC protocols: LPL, MFP with and without filtering of redundant messages that we call MFP-filter and MFP-nofilter, respectively. We compare the gain obtained from avoiding unneeded listening to the preamble (the case of MFP-nofilter) with the gain obtained from both avoiding keeping listening to the preamble and filtering out redundant messages (the case of MFP-filter).

For the routing and application layers, we use simple flooding. Each source periodically generates a message and broadcasts it to other nodes. For each simulation run, we record the number of messages each node has transmitted, correctly received, and the number of collided messages it has observed. Note that we only count the number of distinct received messages, i.e. we do not count redundant copies of the same message received by the same node. Such redundant messages result from the forwarding of the same message by the neighbors of the node. We do not count redundant messages in order to be fair in comparison with MFP-filter that does not receive redundant messages.

We define two particular instants for gathering statistics during simulation: the *first-node*

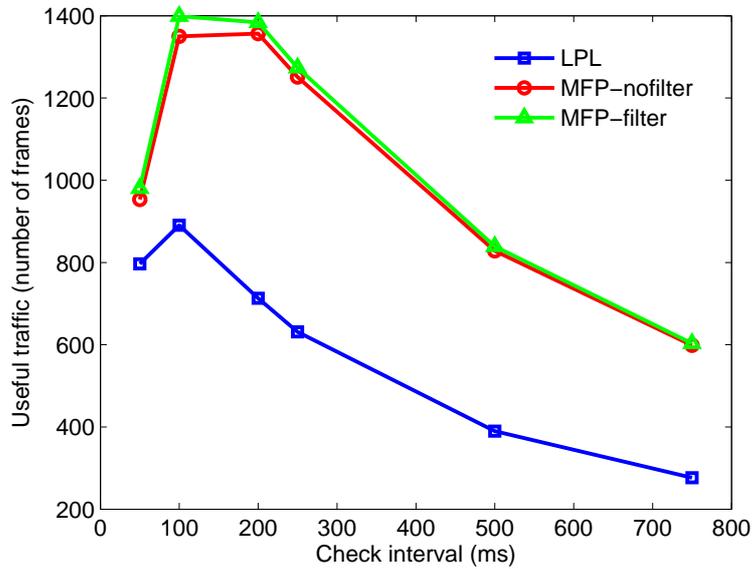


Figure 6.8: The mean lifetime of nodes for various check intervals. It is measured in terms of the number of relevant frames that passed through the node per 1 joule. The lifetime is averaged over all the nodes.

and the *all-nodes* instants. The former corresponds to the instant of the death of the first node whereas the latter is the instant of the death of the last node, i.e. when all the nodes are dead.

We have run extensive simulations for a wide range of parameters. Each point of the following plots is the average over 10 simulation runs.

6.3.1 Network-Scope Evaluation

In the first group of simulation, we set the data frame size to 150 bytes, the traffic generation interval to 100 seconds and we vary the check interval from 50 to 750ms. For each value of the check interval, we measure the lifetime for each node. For this purpose, we introduce the *Information Transport Efficiency* (ITE) that quantifies the amount of information passed through each node during its lifetime. First, we count the number of relevant data frames the node has sent or received. Then, we quantify the ITE as the number of relevant frames passed through the node (or the number of bits) per joule. Note that, the ITE considers only the useful information: redundant broadcast frames do not count. The ITE is more accurate for quantifying the lifetime of a node than just measuring the time it takes a node to run out of energy, because the lifetime of a node also depends on the quantity of information passed through it — extending the lifetime of a node means allowing more information to pass through it per energy unit. Hereafter, the term normalized lifetime refers to the number of relevant frames passed through a node per 1 joule.

Figure 6.8 shows the evolution of the mean lifetime of nodes in function of the check interval. Each point on the curves presents the lifetime averaged over all the nodes and measured at the *all-nodes* instant. Figure 6.8 shows that MFP significantly increases the

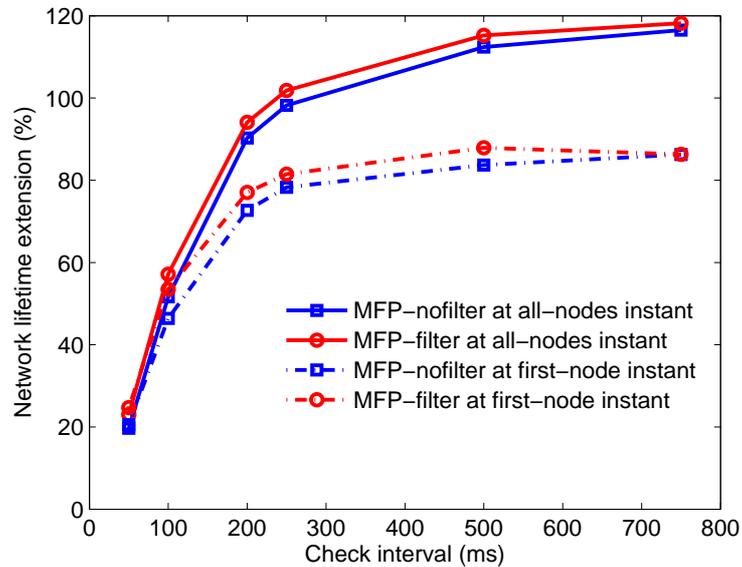


Figure 6.9: The mean lifetime extension of MFP compared to LPL.

mean lifetime of nodes for all check intervals. In addition, filtering irrelevant messages increases the mean lifetime furthermore: the MFP-filter curve is continuously above that of MFP-nofilter. For the parameters of Figure 6.8, filtering redundant messages does not provide significant lifetime increase compared to that of avoiding receiving the preamble — the duration of listening to the preamble is by far longer than that of receiving the data, especially when data frames are quite small and transmitted at high speeds (e.g. 250kb/s). The average lifetime measured at the *first-node* instant (not plotted in the figure) is also maximized when the check interval is between 100ms and 200ms.

Figure 6.9 shows the percentage of lifetime extension measured both at the *first-node* and *all-nodes* instants for MFP-filter and MFP-nofilter compared to LPL. We can see that the gain obtained by using MFP is more important at the *all-nodes* instant, because simulation continues and nodes can improve their energy savings. At the *first-node* instant, the gain corresponds to that of the first node, i.e. when the most vulnerable node exhausts its energy.

Figure 6.10 shows collision rates measured at the *first-node* instant, i.e. from the beginning of the simulation until the first-node instant. Note that after this instant collision rates decrease because the network becomes less dense as nodes start disappearing.

Collision rates measured at the *all-nodes* instant (not plotted in Figure 6.10) exhibit a similar behavior, but with lower rates. Collisions affect nodes lifetime, because in our simulations a node keeps its radio in Receive mode while there is activity on the channel until it decodes a frame correctly or the channel is idle again. If there is a collision, the node spends time in Receive mode listening to the collided frames, which drains energy and thus decreases the lifetime of the node. The energy drained in listening to collisions depends on the transmission duration of the whole message (preamble plus data). The larger the message, the longer the duration of listening to collisions. The preamble length and thus the check interval should be smaller to reduce the amount of energy wasted in listening to

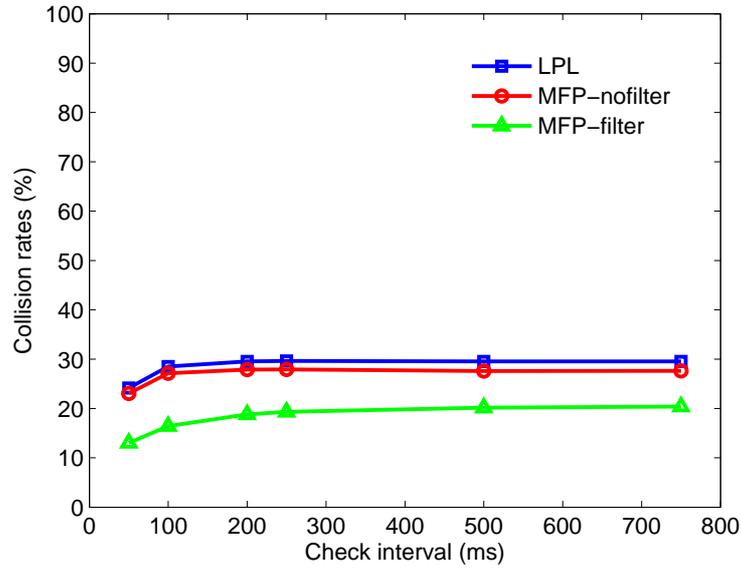


Figure 6.10: The mean collision rates.

collisions.

Collision rates for MFP-nofilter and LPL are fairly similar, however those for MFP-filter are much smaller. Our simulation of MFP does not guarantee that MFP-filter has the same execution sequence as MFP-nofilter and LPL. Figure 6.11 explains why this may happen. Assume that Message 1 is transmitted before Message 2 and the receiver is able to correctly decode a microframe of the Message 1 preamble. This is possible, if the receiver wakes up before Message 2 is transmitted. Then, assume that Message 1 is redundant for the receiver. Thus, MFP-filter will filter it out and switch the radio off during its transmission. Consequently, if the receiver uses MFP-filter protocol, then it will not observe this collision as its radio will be off. However, if the receiver uses MFP-nofilter or LPL, then it will observe a collision as it will wake up to receive the data frame of Message 1, which will be

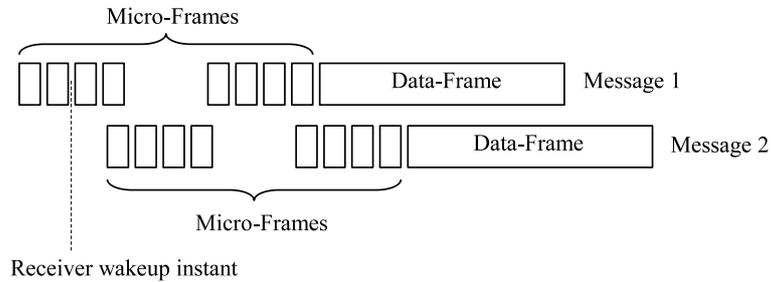


Figure 6.11: Message collision in ns-2 simulation. Messages 1 and 2 are transmitted by nodes that are out of transmission range of each other — the collision occurring at the receiver is caused by the hidden node problem.

Table 6.1: Performance for chosen nodes.

Node Identifier	Degree	Collision	Transmission	Reception	Lifetime extension
Node 1	1	0%	60%	40%	40%
Node 15	8	60%	20%	20%	80%
Node 25	1	0%	50%	50%	80%
Node 0	5	20%	40%	40%	120%
Node 27	7	40%	30%	30%	140%

corrupted by Message 2.

6.3.2 Node-Scope Evaluation

In this section, we present simulation results for some chosen nodes to provide more details on the performance of our protocol. In a random topology, nodes may fall into different categories: low or high degree of connections, low or high traffic load, vulnerable position in the graph or not, etc.

As shown in Table 6.1, lifetime extension ranges from 40% for node 1 to 140% for Node 27. Node 15 is the most vulnerable as it is the first to run out of energy. Node 15 dies before the others because it has the largest number of neighbors (eight) and thus the highest activity rate. Node 25, albeit not as vulnerable as Node 15, presents the same lifetime extension, because it has only one neighbor: Node 15. Therefore, when Node 15 dies, Node 25 becomes isolated and does not forward any traffic. Thus, its lifetime extension remains the same as for Node 15.

Table 6.1 confirms the relation between node degree and lifetime extension. In general, nodes with high degrees such as Node 27 have important lifetime extension. However, when a high degree node experiences high collision rates, which is the case for example for Node 15, the lifetime extension may be lower; for instance, Node 15 has 8 neighbors, but with 60% of its energy drained during collisions, its lifetime extension is only 40%.

For some nodes, the lifetime extension measured at the *all-nodes* instant is smaller than that measured at the *first-node* instant. Such nodes forward a decreasing amount of traffic after the *first-node* instant, so their lifetime extension is smaller. For instance, at the *first-node* instant, Node 15 runs out of energy and Nodes 1, 14, and 29 become disconnected from the network and form an isolated subnetwork. From that instant, the only traffic going through this subnetwork comes from source node 1, which is smaller than the traffic before the *first-node* instant. Thus, the improvement observed afterwards decreases.

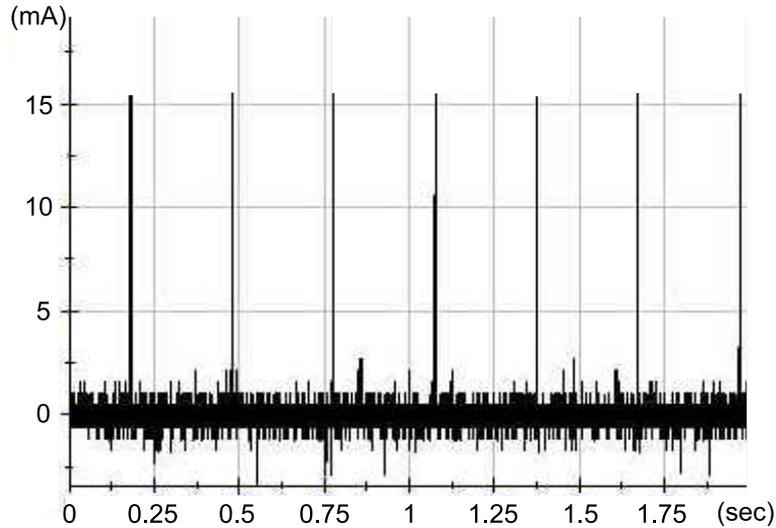
6.4 Implementation

6.4.1 Feasibility of MFP on Existing Radios

To show that MFP is feasible on existing radio modules, we have implemented MFP on the Chipcon CC2500 [11] evaluation board. The module contains a CC2500 chip, a short range low power radio transceiver with a 2.4 GHz modem. The transceiver is controlled by an 8051 low power 8 bits-24MHz microcontroller [96] with 2304B of RAM and 16KB of flash

Table 6.2: *Measured current consumption of CC2500*

Radio (sleep)	900 nA
Radio (idle)	1.5 mA
Radio (transmit)	22 mA
Radio (receive)	14 mA
Microcontroller (active)	8 mA
Microcontroller (idle)	2 mA

**Figure 6.12:** *Current consumption during periodic channel sampling done by the radio module. We have performed these current measurements only on the radio module.*

memory. The radio and the microcontroller communicate via a Serial Peripheral Interface (SPI).

The CC2500 already implements an efficient preamble sampling protocol called WOR (Wake On Radio). WOR is implemented on the radio chip to save the microcontroller the burden of periodically switching the radio between Receive and Sleep modes. This enables programmers to put the microcontroller in Idle (low power) mode while the radio is periodically sampling the channel.

We have implemented a simple flooding application and set the check interval to 300ms⁵.

To evaluate energy savings, we have measured the instantaneous current consumption of the sensor node to identify in which mode it is operating. Table 6.2 shows the current drained for each mode.

Figure 6.12 shows the current consumption during periodic channel sampling. Sampling is done by the radio module so that it is possible to put the microcontroller in Idle (low power) mode while sampling. The current drained to sample the channel is 14mA on average⁶.

⁵CC2500 offers a limited set of check intervals, so we used the value of 300ms.

⁶The average values we measured are slightly larger than the typical values reported in the CC2500 data

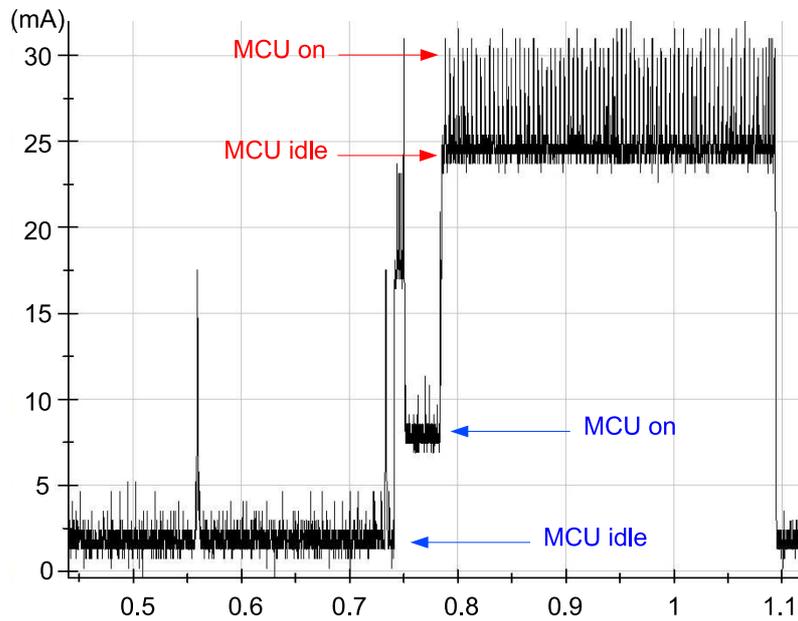


Figure 6.13: Current consumption during frame forwarding using MFP taken at the radio and the microcontroller together.

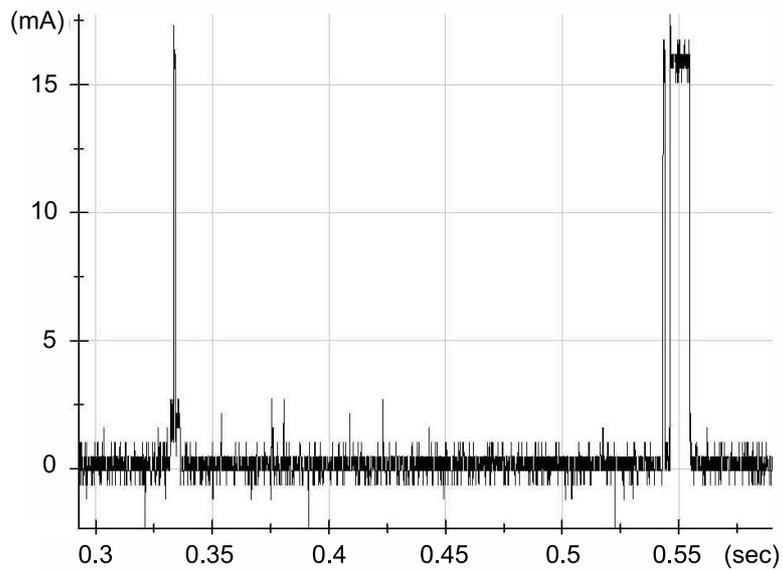


Figure 6.14: Details of a data frame reception. Measurements are performed only on the radio.

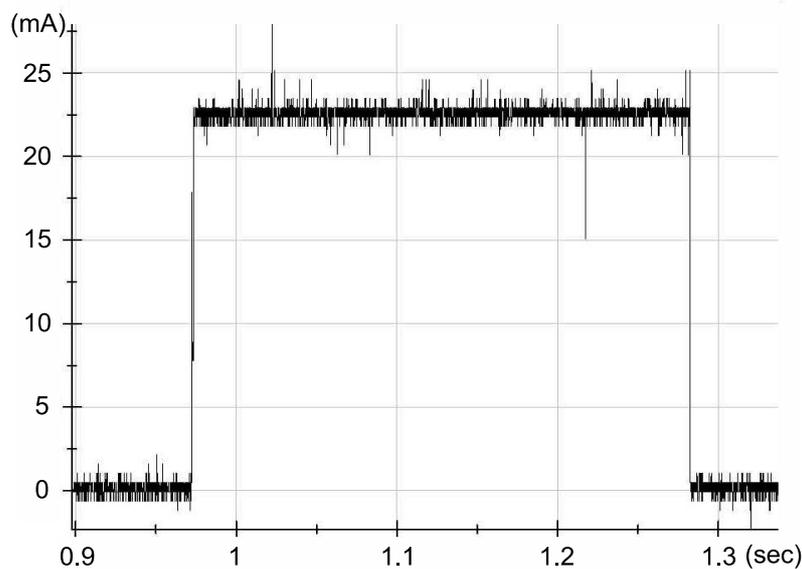


Figure 6.15: *Details of a transmission (microframes and a data frame). Measurements are performed only on the radio.*

Figure 6.13 presents the current consumption during frame forwarding: the node receives a frame to be flooded and forwards it to its neighbors. Initially, the node periodically samples the channel. When the node wakes up to sample the channel at time 0.55s, it receives a microframe from which it learns about the arrival time of the data frame. As the node uses MFP, it switches its radio off until instant 0.74s to save energy. It then wakes up at 0.74 to receive the data frame. Because of potential clock drift, the node actually wakes up slightly ahead of the time for the data frame. During this wakeup, the node receives a second microframe and then the data frame. Figure 6.14 shows some details: the reception of two microframes and the data frame. Note that time scales are different on each graph.

Our implementation has brought into focus the problem of the clock drift. This practical detail does not look important unless one tries to implement the MFP protocol. When a node receives a microframe indicating that the forthcoming data frame should be received, it calculates from the sequence number the sleep duration until the data frame. The clock needs to be precise enough to switch on the radio right on time to receive the data frame. The node should wake up early enough to make up for the clock drift and correctly receive the data frame. The accuracy of the oscillator used by the node determines the amount of the extra time the node spends in receive mode before the data frame arrives. There is a trade-off to make between using a high precision oscillator such as quartz crystal (ranging from 1ppm to 100ppm) or a lower precision oscillator such as a RC (e.g. ≈ 15000 ppm [96]). A RC oscillator is less expensive and occupies less space than a quartz crystal. In our case, we use RC oscillator as it is integrated in the module.

After the reception of the data frame, the node should forward it to its neighbors. Before each message transmission (microframes plus data), the node chooses a random backoff

sheet. In the figure, peaks exceed 14mA because we measure instantaneous current.

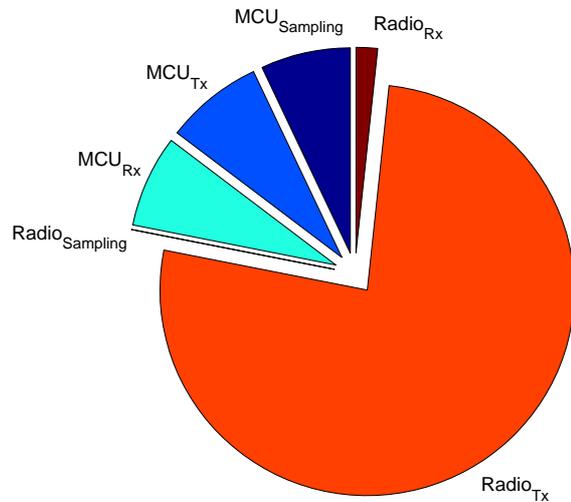


Figure 6.16: *Distribution of the energy consumption of both the radio and the microcontroller for the different MFP operations.*

to avoid collisions due to simultaneous access to the channel. Note that contrary to the IEEE 802.11 DCF [43], but similarly to the IEEE 802.15.4 CAP [38], the channel is not continuously sensed during the backoff. The radio is switched off during the whole backoff period to save energy. When the backoff timer expires, the node performs carrier sense and transmits at once if the channel is free. Otherwise, if the channel is busy, the node re-executes the backoff procedure.

The backoff appears in Figure 6.13 during the interval from 0.755s to 0.78s. The current drained during the backoff procedure is higher by 8mA, because we put the microcontroller in active mode during the backoff to show the difference between the microcontroller's active and idle modes on the same figure. After carrier sense, the node begins to transmit by first sending a series of microframes and then the data frame. During the transmission phase, the current drained by the node (radio plus microcontroller) oscillates between 24mA and 30mA. The value of 24mA corresponds to the radio in Transmit mode (22mA) (see Figure. 6.15) and the microcontroller in Idle mode (2mA), whereas the value of 30 corresponds to the radio in Transmit mode and the microcontroller in Active mode (8mA). In our implementation, we reduce the time during which the microcontroller is in Active mode; for example, the microcontroller fills in the transmission buffer of the radio with frames to be transmitted and goes back to Idle mode when the radio's transmission buffer is full. At the same time, the radio is reading its transmission buffer and transmitting frames. As the microcontroller fills the buffer (6MHz SPI) quicker than the radio is able to transmit (bandwidth 250kb/s), the microcontroller can go back to Idle mode while the radio is still transmitting frames.

6.4.2 Microcontroller Overhead

As shown in the previous section, the energy efficiency of MFP depends on the power consumed by the radio and on that consumed by the microcontroller. In this section, we determine the energy consumption distribution of these two components in the three operation modes of MFP: sampling, transmission, and reception. The transmission of a frame requires that the radio remains in transmit mode that consumes the highest amount of current (22mA) for a long time (microframes plus the data). Therefore, the radio transmission is the dominant energy consuming operation as shown in Figure 6.16. The radio reception operation is made very small by our MFP protocol that saves nodes the reception of preamble until the data. The radio sampling is the smallest energy consuming operation, which is expected because it is the main objective of preamble sampling protocols.

For the microcontroller, the energy consumption of the three considered operations (sampling, reception, and transmission) are almost equal, as shown in Figure 6.16. The reason is that our implementation puts the microcontroller in Idle mode whenever possible, therefore the energy consumption of these operations is dominated by the consumption of the microcontroller's Idle mode. In the next section, we show that reducing the energy consumption of the microcontroller's Idle mode substantially increases energy saving and thus nodes lifetime.

6.4.3 Possible Improvements

The MFP protocol has three major states: channel sampling, transmission, and reception. The implementation of MFP on the CC2500 drastically minimizes the microcontroller overhead by putting it in Idle mode whenever possible.

For the channel sampling operation, the microcontroller goes to Idle mode (or to any other low power mode) while the radio is performing periodic wake up according to a predefined check interval. Thus, the microcontroller does not consume any more energy in channel sampling mode than in Idle mode.

The reception operation starts when the radio receives a frame while periodically sampling the channel. Upon this reception, the radio wakes the microcontroller up to process the received frame. If the microcontroller finds a microframe, then it checks whether the subsequent data is relevant or not. If so, the microcontroller goes back to Idle and wakes up later only to receive the data.

For the transmission, the microcontroller constructs microframes and sends them to the radio's transmission buffer. At the same time, the radio reads data from the buffer and sends them to the air interface. As the microcontroller is faster than the radio, the buffer may be full during some periods. When the buffer is full, the microcontroller goes back to Idle mode. When the buffer is almost empty, the radio wakes the microcontroller up so that it continues to fill the buffer with frames to be transmitted.

In all these modes, the microcontroller is in low power mode most of the time especially when the traffic load is low. Figure 6.17 shows that reducing the current drained in the Active mode only offers a negligible gain, whereas reducing that of the Idle mode significantly increases the lifetime. Therefore, it is better for MFP to use microcontrollers with a minimum power consumption Idle mode. This result emphasises the negligible overhead of calculating microframes by the microcontroller, therefore there is no need to add dedicated

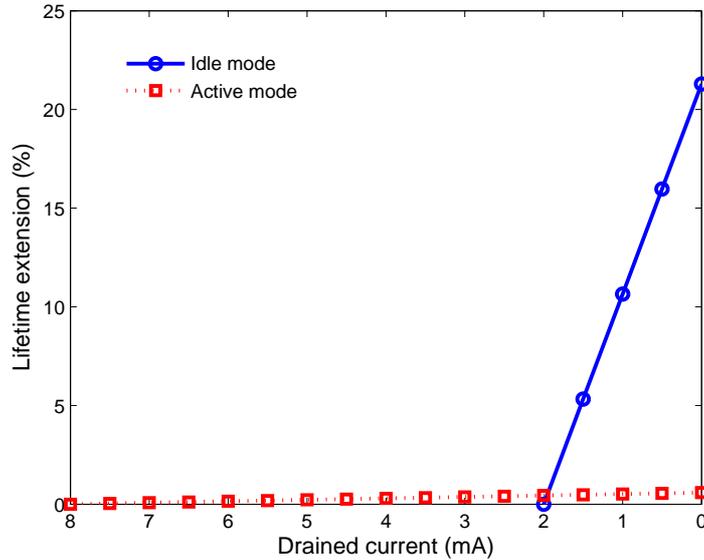


Figure 6.17: *The percentage of lifetime extension in function of different values for the current drained by the microcontroller in Idle and Active modes. Reducing the current drained in the Active mode only offers a negligible gain, whereas reducing that of the Idle mode significantly increases the lifetime.*

hardware for calculating microframes.

6.5 MFP Compared with Other Protocols

In this section, we focus on contributions related to preamble sampling. We show that MFP can replace these contributions in some situations and can be used jointly with them in other situations. In both cases, MFP increases the amount of energy savings. For more detailed description of the protocols discussed hereafter refer to Chapter 2.

BMAC [41] proposes an outlier detection based technique to improve the accuracy of CCA (Clear Channel Assessment). An accurate CCA has two major benefits. First, it reduces the number of false assessments (e.g. the channel is assessed to be busy while it is actually clear), which increases channel utilization and thus the network throughput. Second, it allows a node to accurately detect whether the channel is still active during a presumed preamble reception, which reduces the duration of receiving false preambles and thus increases energy savings.

The accurate CCA operation of BMAC is compatible with the two versions of MFP: persistent MFP (presented in this chapter) and non persistent MFP (presented in the next chapter). In the persistent MFP, in which a node persists in receiving until it receives a microframe or the channel is back to idle, an accurate CCA allows a node to detect when the channel is back to idle so that it stops receiving at the right time. In non-persistent MFP, in which a node does not persist in reception but gives up if it fails to receive a microframe within a certain timeout value, an accurate CCA only increases channel utilization as it is

not needed to detect false preambles.

WiseMAC [25] and SCP [24] reduce the preamble length by using short preamble for unicast frames. As they still use preambles, albeit not always of a full-length, microframes can replace the continuous preambles, thus achieving further augment energy savings.

Protocols that use preambles split into frames with a gap between consecutive frames, such as STEM-B [28], CSMA-MPS [27], TICER [50], WOR [11] and X-MAC [26], have the advantage of not always needing the full-length preamble; in the case of unicast transmissions, the receiver sends the ACK in the gap between the frames, thus stopping the preamble transmission. However, in very lightly loaded networks, these protocols do not guarantee optimal energy savings, because they increase idle listening at receivers. When there is a gap between frames, a receiver should remain in Receive mode for a duration that is larger than the gap to sample the channel. The sampling duration therefore increases and nodes waste more energy in sampling. MFP avoids this long sampling duration by sending microframes without gaps. MFP is more suitable for low data rate networks.

Z-MAC [52] is a hybrid protocol that combines the strengths of CSMA and TDMA. Under low contention, Z-MAC switches to CSMA to achieve high channel utilization and low delays. Under high contention, Z-MAC switches to TDMA to achieve high channel utilization, fairness, and less collisions. Z-MAC is implemented in TinyOs [7] on top of B-MAC. Therefore, it can be used with MFP that is compatible with B-MAC. MFP improves energy savings of B-MAC, therefore it may do the same for Z-MAC.

6.6 Conclusions

Protocols based on preamble sampling techniques are being increasingly used in wireless sensor networks because of their significant energy savings compared to other protocols. In this chapter, we have shown that energy consumption can be further reduced by reducing irrelevant receptions. Our technique, called MFP (Micro Frame Preamble), replaces the continuous long preamble by a series of micro-frames containing an indicator of the data frame contents and a sequence number. The information enables a node to switch off the radio and wake up only for the reception of a relevant data frame. MFP is not another MAC protocol for sensor networks; rather, it is a generic technique that can be used under various preamble protocols including, WiseMAC, LPL, Z-MAC, SCP and others.

In this chapter, we have presented an extensive evaluation of MFP. We have derived analytical formula for lifetime extension obtained with MFP, and performed ns-2 simulations to include other parameters, such as collisions, into account. We have shown that MFP can be successfully implemented on existing radio modules, such as the CC2500, without hardware modifications. We have also shown that the overhead of computing and passing microframes to the radio chip is negligible so that MFP can be implemented on a microcontroller with a limited processing power.

In the next chapter, we extend our study to various schemes of frame preamble MAC protocols in which the preamble can be a series of micro frames or replicas of the data frames. For these protocols, we study the relation between the energy drained over a link and the corresponding reliability according to channel errors.

7 Evaluating Link Cost and Reliability of Frame Preamble Protocols

The wireless communication medium used in sensor networks is not error-free. It may be subject to degradations that may corrupt some transmitted frames so that they cannot be correctly decoded by potential receivers. These transmission errors affect both the energy consumption and the reliability of wireless links. There is a dual relation between energy consumption and reliability. On one hand, higher reliability often requires more sophisticated modulation and coding schemes, or stronger transmission power, which increases energy consumption. On the other hand, higher reliability increases the robustness of transmissions thereby less retransmissions are required, which reduces energy consumption.

In this chapter, we study the relation between the energy drained by some variants of MAC protocols and their corresponding reliability. We consider variants of Frame Preamble MAC protocols, which are preamble sampling based protocols with the preamble replaced by a series of frames. Our aim is to determine the variant that achieves the minimum energy consumption according to a predetermined reliability threshold.

We organize the remainder of this chapter as follows. First, we describe the variants of Frame Preamble MAC protocols used in this chapter (Section 7.1). Next, we propose a generic model that provides the reliability, transmission, and reception costs for the Frame Preamble MAC (Section 7.2). Then, we use this generic model to evaluate the link cost and the reliability of four variants studied in this chapter and we discuss the results (Section 7.3). Finally, we give some concluding remarks (Section 7.4).

7.1 Frame Preamble MAC Protocols

We propose four variants of Frame Preamble MAC protocols based on different strategies used by nodes to transmit and receive frames. They result from a combination of two transmission: MFP (Micro Frame Preamble) or DFP (Data Frame Preamble), and two reception schemes: Persistent or Non Persistent.

7.1.1 Transmission Schemes: MFP vs. DFP

Frames used in the Frame Preamble MAC can be either microframes (MFP) carrying some information about the data frame or simply duplicated copies of the data frame itself (DFP)—Data Frame Preamble. As MFP has been described in the previous chapter, we restrict the following description to DFP.

In DFP, the frames transmitted instead of the preamble are duplicated copies of the data frames. The advantage of DFP is that the node that wakes up to check the channel immediately receives the data, thus it does not need to wake up again to receive the data frame as it is the case in MFP. DFP also presents another advantage: duplicating the

7 Evaluating Link Cost and Reliability of Frame Preamble Protocols

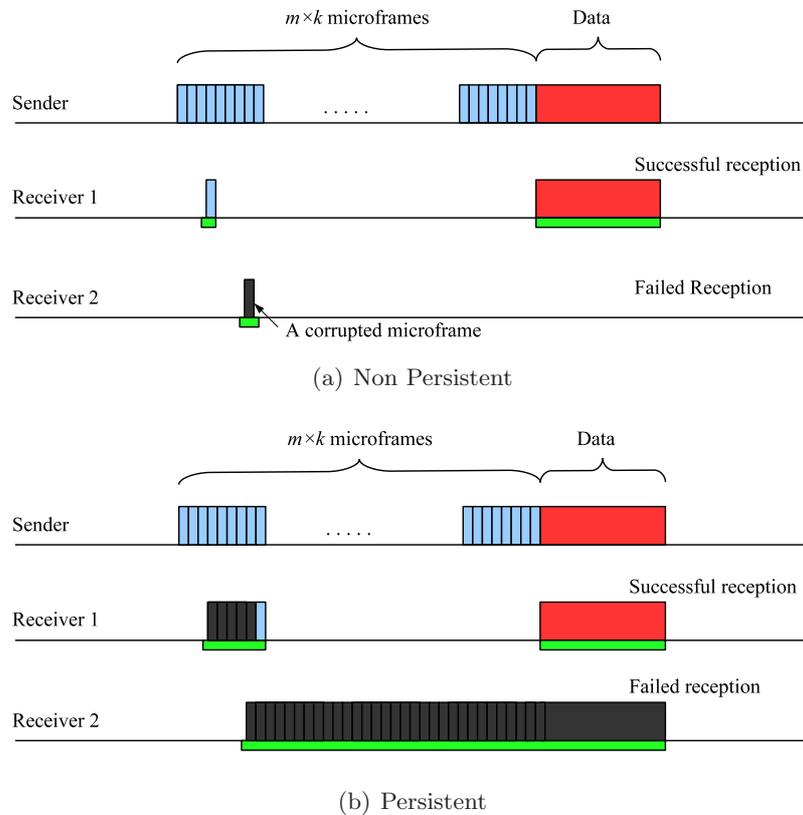


Figure 7.1: MFP

data in preamble frames increases the reliability of the transmission (we show this effect in Section 7.3.5). However, in DFP the node cannot avoid receiving irrelevant data, which may consume non-negligible energy if the data frames are large or when they are transmitted at a low bandwidth.

7.1.2 Reception Schemes: Persistent vs. Non Persistent

In Frame Preamble MAC protocols, nodes wake up periodically each check interval to sense the state of the channel. The time needed for this operation is the sampling time which duration depends on the preamble transmission scheme (MFP or DFP). The sampling time should be large enough so that it is possible for a node to decode the information being transmitted on the channel. For example, if the transmission scheme is MFP, then the sampling time is at least equal to one microframe transmission time, because a node needs to correctly decode a microframe to know when the data frame will be transmitted and whether it is relevant. In general, a node needs more than the minimum sampling time to correctly decode a microframe. The sampling time depends on the instant of node wakeup and the quality of the radio link. If we assume that radio links are perfect then the maximum sampling time needed is equal to twice the microframe transmission time. This happens when the node fails to receive a microframe because it has missed its first bit. In this case, the node should keep receiving until it decodes the subsequent microframe. However, if the

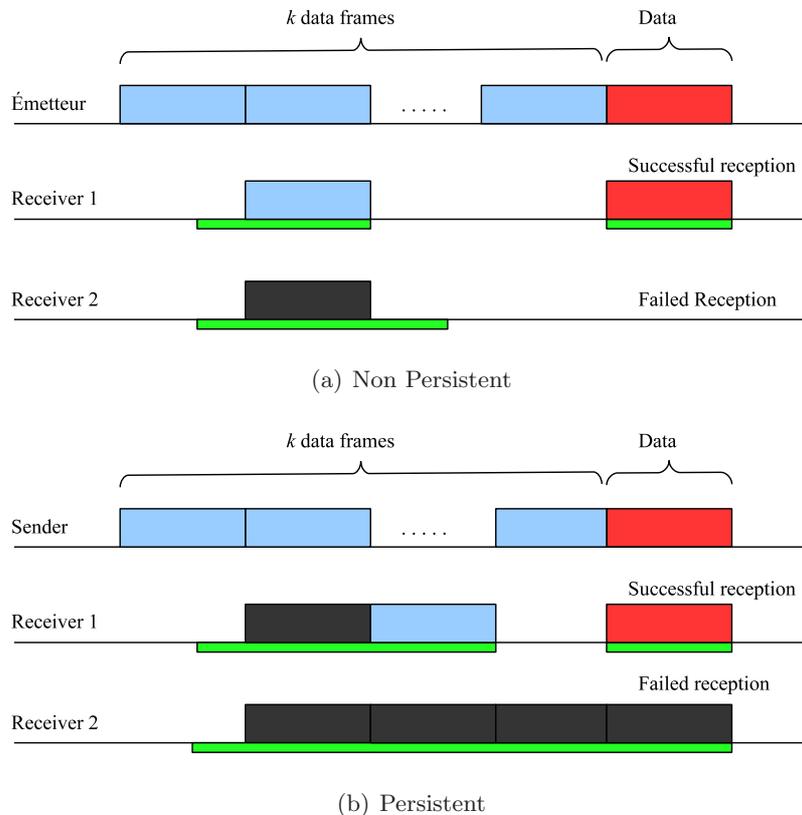


Figure 7.2: DFP

radio link is not perfect, then transmission errors may occur and thus the node may even fail to receive the subsequent microframe. In this case, the node has two options: either (i) to be *persistent* and continue receiving until it decodes a frame or the channel becomes idle, or (ii) to be *non-persistent* and stop receiving after a timeout value. Figure 7.1 shows an example of persistent MFP (Figure 7.1(b)) and non-persistent MFP (Figure 7.1(a)). Note that the notion of persistence also applies to DFP as well: we therefore consider two other protocols: persistent DFP (Figure 7.2(b)) and non-persistent DFP (Figure 7.2(a)).

Our intuition is that the non persistent methods should be used in channels with bursty errors or with high error rates. In bursty channels, the node goes back to sleep to avoid keeping receiving during the burst of errors and wakes up later on to sample the channel again. In channels with high error rates, the non persistent method saves the node the energy of keeping receiving without success as the probability of a correct reception is low. The persistent method is efficient in channels with low error rates. Persisting in reception under these circumstances saves the transmitter the cost of retransmitting.

7.2 System Model

7.2.1 Problem Statement and Assumptions

We consider a wireless link between two nodes. We want to find the energy cost of transmitting one data frame over this link and to estimate the corresponding reliability. The energy cost of a link is the amount of energy drained both at the transmitter and the receiver. The reliability is the probability that both the receiver correctly decodes the data frame. For the analysis described in this chapter, we assume that the receiver and the transmitter use the same variant of the Frame Preamble MAC, i.e. both use pDFP (persistent DFP), or both use npMFP (non persistent MDP), etc. Our aim is to identify the best variant that consumes the least amount of energy while ensuring a certain level of reliability.

We assume a Binary Symmetric Channel (BSC) in which each bit has a constant and independent error probability. We call p the probability that a microframe is corrupted. We assume that a microframe has a unit size and its transmission time has a unit duration. For the sake of simplicity, we express our results in function of these units. We assume that the size of data frames is m times larger than that of microframes and there are $m \times k$ microframes in the preamble, where k is some constant. As we use microframes as unit sizes and unit durations, the transmission of a data frame also has the duration of m and the check interval the duration of mk (see also Figure 7.1). We assume that DFP protocols send k data frames in the preamble (see Figure 7.2). Constant k enables us to relate two types of protocols and to compare different variants.

7.2.2 Reliability

In Frame Preamble protocols, once a node receives a data frame, it sends an ACK message back to the transmitter to acknowledge a successful reception. If the transmitter does not receive the ACK, then it retransmits again until it receives the ACK or the maximum number of transmissions is reached. Each retransmission includes the whole Frame Preamble plus the data. We call n the maximum number of transmissions and p_f the probability of a failed single transmission. Therefore, reliability R , which is the probability of a successful communication within n single transmissions is:

$$R = 1 - p_f^n. \quad (7.1)$$

7.2.3 Transmission Cost

The energy drained in transmission is proportional to the amount of time the transmitter spends in transmit mode. It is also proportional to the current and voltage, but for the sake of simplicity, we assume these to be constant. We distinguish between a *transmission* and a *single transmission*. A single transmission involves only the preamble and the data, whereas a transmission also includes single retransmissions. We call T the duration of a single transmission:

$$T = mk + m \quad (7.2)$$

We call T_{tx} the average transmission duration. We have:

$$T_{tx} = (1 - p_f)T + p_f(1 - p_f)2T + \dots + p_f^{n-2}(1 - p_f)(n - 1)T + p_f^{n-1}nT$$

$$\begin{aligned}
&= (1 - p_f) \sum_{i=1}^{n-1} p_f^{i-1} iT + p_f^{n-1} nT \\
&= \frac{1 - p_f^n}{1 - p_f} T.
\end{aligned} \tag{7.3}$$

7.2.4 Reception Cost

We follow the same methodology as in Section 7.2.3 to derive the average time the receiver spends in receive mode. Let S (resp. F) be a random variable that expresses the time the receiver spends in receive mode in case of a successful (resp. failed) single transmission. Therefore, the reception duration T_{rx} is

$$\begin{aligned}
T_{rx} &= (1 - p_f)S + (1 - p_f)p_f[F + S] + \cdots + (1 - p_f)p_f^{n-1}[(n - 1)F + S] + p_f^n nF \\
&= (1 - p_f) \left(\sum_{i=0}^{n-1} p_f^i [iF + S] \right) + p_f^n nF \\
&= \frac{1 - p_f^n}{1 - p_f} [p_f F + (1 - p_f)S].
\end{aligned} \tag{7.4}$$

7.3 Evaluation

For the evaluation of Frame Preamble protocols, we find the values of p_f , F , and S for each protocol variant: non-persistent-DFP, non-persistent-MFP, persistent-DFP, and persistent-MFP.

7.3.1 Non-Persistent DFP

In non-persistent DFP (npDFP), the timeout value is twice the data transmission time. Thus, to correctly receive a single transmission, the receiver must correctly decode the data frame following its wakeup instant. Let us call q the probability that a data frame is corrupted:

$$q = 1 - (1 - p)^m. \tag{7.5}$$

Therefore,

$$p_f = q. \tag{7.6}$$

In DFP, the receiver may wake up at any instant during the transmission of preamble data frames. If the receiver wakes up and misses the first bit of a preamble data frame, then it should keep receiving until it meets the first bit of the next preamble data frame. This time is equal to U_m , where U_m is a uniform random variable in $[0, m[$. When the receiver detects the first bit of this data frame, it continues receiving until it receives the whole data frame which duration is m . Therefore, S , the average duration of a successful reception within a single transmission is:

$$S = U_m + m. \tag{7.7}$$

However, in a failed single transmission, reception duration F depends on the wakeup instant of the receiver. If the receiver wakes up during the last preamble data frame, then it fails to decode the subsequent data frame. In this case, the receiver goes back to sleep before the timeout expires, because the channel is back to idle before. In this case, F is equal to $U_m + m$. However, if the receiver wakes up before the last preamble data frame, then it goes back to sleep again when the timeout expires. As the duration of the timeout in npDFP is $2m$, F is equal to $2m$ in this case. The probability that the receiver wakes up during the last preamble data frame is $1/k$. Therefore, we have:

$$F = \frac{k-1}{k} \times 2m + \frac{1}{k} \times (U_m + m). \quad (7.8)$$

7.3.2 Non-persistent MFP

In non-persistent MFP (npMFP), a successful reception within a single transmission requires that the receiver correctly decode the microframe following its wakeup instant during the preamble and correctly decode the data frame transmitted after the preamble. Therefore, p_f is as follows:

$$\begin{aligned} p_f &= 1 - (1-p)(1-q) \\ &= 1 - (1-p)^{m+1}. \end{aligned} \quad (7.9)$$

For the sake of simplicity, we do not consider the case in which the receiver wakes up during the last microframe. This assumption simplifies the analysis while having very negligible effects on the results as this case happens rarely (e.g. if the preamble contains 100 microframes, then this case occurs with a frequency of 1% on the average). Thus, the average duration of a successful reception within a single transmission S is equal to the duration of receiving a correct microframe plus that of receiving a correct data frame. Therefore,

$$S = U_1 + 1 + m. \quad (7.10)$$

where $(U_1 + 1)$ is the average reception duration needed to receive one microframe. U_1 is a uniform random variable in $[0, 1[$.

A single transmission fails, because either the receiver fails to receive a microframe in the preamble so that it does not wake up to catch the data frame, or the receiver correctly decodes a microframe, but fails to correctly receive the data. As the probability of the first case is p , we have:

$$F = p \times 2 + (1-p) \times (U_1 + 1 + m). \quad (7.11)$$

where 2 is the timeout duration, which is equal to twice the duration of two consecutive microframes in the case of MFP.

7.3.3 Persistent DFP

In persistent DFP (pDFP), a single transmission fails when the receiver fails to receive all data frames before the channel is back to idle. This includes preamble data frames and the data frame transmitted after the preamble. As its wakeup instant is random, the receiver

may miss the reception of j data frames, where j is in $1, \dots, k$. For example, if the receiver wakes up during the first preamble data frame, then it may keep listening during all the $k - 1$ subsequent preamble data frames plus the subsequent data frame. In this case, the number of missed data frames is equal to k . As the probability that the receiver wakes up during the transmission of any preamble frame is $1/k$, the probability p_f of a failed single transmission is:

$$\begin{aligned} p_f &= \frac{1}{k}q^k + \frac{1}{k}q^{k-1} + \dots + \frac{1}{k}q \\ &= \frac{q}{k} \left(\frac{1 - q^k}{1 - q} \right). \end{aligned} \quad (7.12)$$

To find the distributions of F and S , we introduce $X \in \{0, \dots, k - 1\}$, a discrete random variable that expresses the number of received corrupted preamble data frames. We have:

$$X = (X|_{failure})p_f + (X|_{success})(1 - p_f). \quad (7.13)$$

where, $X|_{failure}$ (resp. $X|_{success}$) is a discrete random variable that expresses the number of received corrupted preamble data frames knowing that the single transmission failed (resp. succeeded). Therefore, F and S are as follows:

$$F = U_m + m \times (X|_{failure}) + m. \quad (7.14)$$

$$S = U_m + m \times (X|_{success}) + m. \quad (7.15)$$

To calculate $P[X = j]$ for $j = 0, \dots, k - 1$, we use the following relation:

$$P[X = j] = P[X \geq j] - P[X \geq j + 1]. \quad (7.16)$$

To find $P[X \geq j]$, we introduce $Z \in \{1, \dots, k\}$, a random variable that expresses the position of the preamble data frame during which the receiver wakes up. For example, if the receiver wakes up during the transmission of the first preamble data frame, which has position 1, then $Z = 1$. The random variable Z is uniform in $\{1, \dots, k\}$, i.e. $P[Z = j] = 1/k$. Therefore, we have:

$$\begin{aligned} P[X \geq j] &= \sum_{i=1}^k P[X \geq j | Z = i] P[Z = i] \\ &= \underbrace{\frac{1}{k}q^j + \dots + \frac{1}{k}q^j}_{\text{if the receiver wakes up before position } k-j} + \underbrace{\frac{1}{k}0 + \dots + \frac{1}{k}0}_{\text{otherwise}} \\ &= \frac{k-j}{k}q^j. \end{aligned} \quad (7.17)$$

Consequently,

$$\begin{aligned} P[X = j] &= \frac{k-j}{k}q^j - \frac{k-(j+1)}{k}q^{j+1} \\ &= \frac{q^j}{k} \left[k(1-q) + q - j(1-q) \right]. \end{aligned} \quad (7.18)$$

7 Evaluating Link Cost and Reliability of Frame Preamble Protocols

We use the random variable X which distribution is calculated in Eq. (7.18) to express both of $X|_{failure}$ and $X|_{success}$.

The random variable $X|_{failure}$ is uniform in $\{0, \dots, k-1\}$, because the probability that the receiver keeps listening during j preamble data frames knowing that the single transmission fails is exactly the probability that the receiver wakes up during preamble data frame of position $j+1$, which is equal to $1/k$. Therefore, $P[X|_{failure} = j] = 1/k$.

The random variable $X|_{success}$ can be deduced by the following.

$$X|_{success} = \frac{X - (X|_{failure})p_f}{1 - p_f}. \quad (7.19)$$

7.3.4 Persistent MFP

The probability of a successful single transmission in case of persistent MFP (pMFP) depends only on the data frame that follows the microframes. Therefore, p_f is as follows:

$$p_f = q. \quad (7.20)$$

To calculate F and S , we introduce Y , a random variable that expresses the number of received microframes, corrupted or not: $Y \in \{0, \dots, mk-1\}$. Therefore, F and S can be expressed as follows:

$$F = U_1 + Y + m. \quad (7.21)$$

$$S = U_1 + Y + m. \quad (7.22)$$

U_1 represents the part of the microframe the receiver has to listen to until the beginning of the next microframe and m is the duration of the data frame. Note that Y does not depend on the success of the reception, because the latter only depends on the data frame. Therefore, $F = S$ as shown in the two equations above. To find the distribution of Y , we use the following relation:

$$P[Y = j] = P[Y \geq j] - P[Y \geq j+1]. \quad (7.23)$$

To calculate $P[Y \geq j]$, we follow the same methodology used in Section 7.3.3. We introduce Z that expresses the position of the microframe during which the receiver wakes up. Z is uniform in $\{1, \dots, mk\}$, then $P[Z = j] = 1/mk$.

We have,

$$\begin{aligned} P[Y \geq j] &= \sum_{i=1}^k P[Y \geq j | Z = i] P[Z = i] \\ &= \underbrace{\frac{1}{mk} p^{j-1} + \dots + \frac{1}{mk} p^{j-1}}_{\text{if the receiver wakes up before position } mk-j} + \underbrace{\frac{1}{mk} 0 + \dots + \frac{1}{mk} 0}_{\text{otherwise}} \\ &= \frac{mk-j}{mk} p^{j-1}. \end{aligned} \quad (7.24)$$

Therefore,

$$P[Y = j] = \frac{1}{mk} \left[(mk-j)p^{j-1} - (mk-j-1)p^j \right]. \quad (7.25)$$

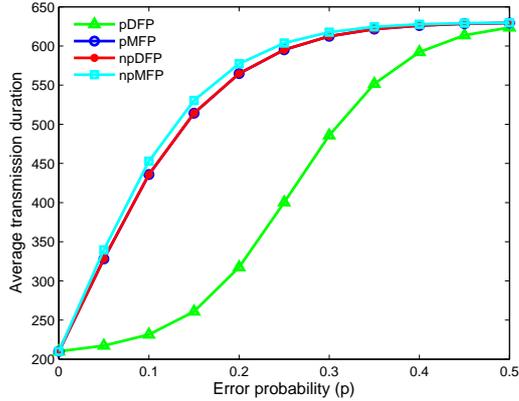


Figure 7.3: Average transmission duration

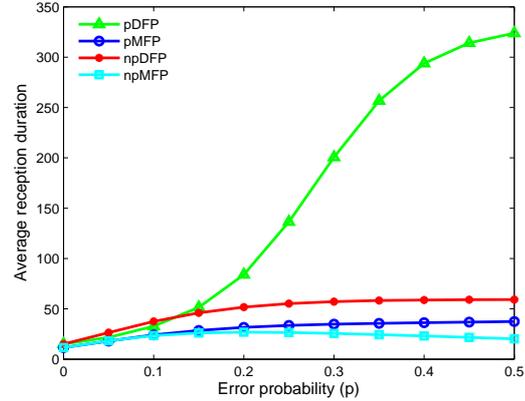


Figure 7.4: Average reception duration

7.3.5 Numerical Examples

In this section, we analyze the results obtained above by plotting the cost of a link and the corresponding reliability in function of transmission error rates. The link cost is determined by the sum of transmission and reception durations expressed in microframe transmission duration unit time and the transmission errors are expressed in microframe error rate.

To obtain the results in Figure 7.3 — Figure 7.6, we have used the following parameters: the maximum number of retransmissions n is 3, the data frame size is 10 times larger than one microframe size (i.e. $m = 10$), the check interval is 200 (i.e. $k = 20$), and the microframe error rate p varies from 0 to 0.5.

As shown in Figure 7.3, pMFP and npDFP have equal average transmission durations, because in both cases the transmitter retransmits the same amount of times. The probability of a successful single transmission, equal to the probability of a correct reception of a data frame is the same in both cases. Likewise, p-MFP and npDFP also have the same reliability, which is confirmed in Figure 7.6.

The average transmission duration of npMFP is slightly larger than that of the other variants (see Figure 7.3), because on the average the transmitter retransmits more times in npMFP than in the other variants. The probability of a single transmission failure in npMFP is larger than that in the other variants, because it does not only depend on a correct reception of a dataframe, but also on a correct reception of a microframe in the preamble.

In Figure 7.3, we also show that pDFP has the shortest average transmission duration, which is expected because pDFP has the lowest probability that a single transmission fails. Figure 7.6 confirms this result as it shows that pDFP has the highest reliability.

In Figure 7.4, we show that npMFP has the shortest average reception duration. For low error probabilities from $p = 0$ to $p = 0.2$, this duration increases when p increases, because in general, for these error rates the receiver correctly decodes a microframe, but fails to decode a data frame. However, for higher error probabilities, from $p = 0.3$ to $p = 0.5$, this duration decreases when p increases, because in this case the receiver mostly fails to decode a correct microframe and then does not wake up later on to listen for the data frame.

7 Evaluating Link Cost and Reliability of Frame Preamble Protocols

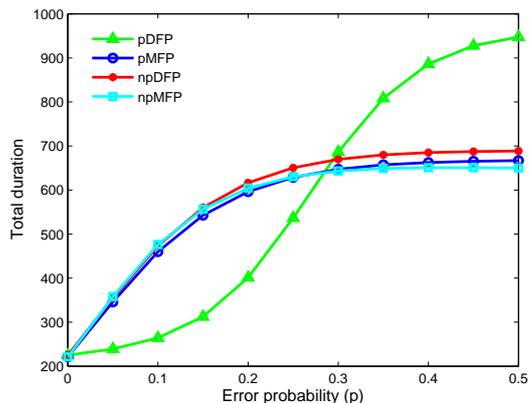


Figure 7.5: Average total duration

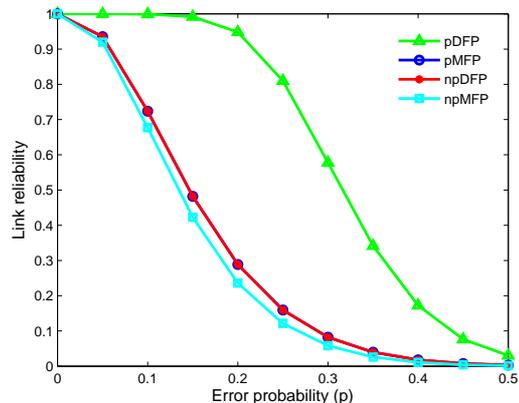


Figure 7.6: Link reliability

In Figure 7.4, we also show that the average reception duration in pDFP increases when the error rate increases, because the receiver has to listen to more data frames in the preamble to correctly decode one of them. Note that this duration is limited on the average to half of the preamble plus the data frame. This limit is reached when the error rate is extremely high, i.e. when p is close to 1, in which case (not shown in the figures) the average reception duration for pMFP is also the same as that for pDFP. The average reception duration in pMFP increases when p increases, but less sharply than that in pDFP, because the receiver in pMFP has a larger probability to decode a microframe in the preamble, thereby it can switch its radio off in the meantime to wake up only to receive the data frame.

In Figure 7.5 and Figure 7.6, we show the relation between the link cost and its corresponding reliability: an increased reliability results in a reduced link cost. Increasing the reliability is often realized by the persistent method, therefore it increases the average reception duration. However, it also reduces the transmission duration as less retransmissions are required. As the transmission cost is the dominant part of a link cost, avoiding retransmission is more energy saving than reducing reception to the preamble. However, this result may not apply when the error rate is very large, because the probability of correctly receiving a frame becomes very low. In this case, the non persistent methods are better, because there is no gain from persisting in reception that likely ends up in a failure.

The figures show that pDFP is the best variants when the channel contains errors. However when the channel is error free or have very reduced error rates, most of the communications succeed within a single transmission. In this case, npMFP is better because it has the smallest average reception duration.

The analysis presented in this chapter is carried out over one link and for a simple BSC channel. Thus, it should not be seen as a final result, but as a step toward understanding the properties of preamble sampling protocols. It shows that channel errors (and by analogy collisions) may strongly influence the performance of preamble sampling protocols. To obtain more significant evaluation, more realistic models of channel errors should be taken into consideration.

7.4 Conclusion

MAC protocols based on preamble sampling offer significant energy savings for low data rate sensor network. In the previous chapter, we have shown that replacing the continuous preamble with a series of microframes further improves energy savings. In this chapter, we have generalized this idea by proposing the Frame Preamble MAC, in which the preamble may be a series of any type of frames. By combining two transmission and two reception strategies, we have considered four variants of Frame Preamble MAC protocols: persistent MFP, non persistent MFP, persistent DFP, and non persistent DFP. For these variants, we have investigated the relationship between energy cost and communication reliability over a wireless link by assuming a simple binary symmetric channel error model. We have provided a comprehensive mathematical analysis that derives the cost of transmission, reception, and the corresponding reliability according to transmission error rates.

Although the analysis presented in this chapter is carried out on one link with a simple BSC channel and should be further improved, the obtained results help to better understand the main properties of preamble sampling protocols. For example, our results show the increasing cost of transmission failures that should be avoided as much as possible. Moreover, some variants are better than the others for a given transmission error rate, which brings the idea of using an adaptive protocol that changes to the best variant according to the observed error rate.

The energy efficient MAC protocols studied in this and the previous chapters only have a local scope as they consider maximizing the lifetime of each node independently of the others. Therefore, there is a need for combining them with energy efficient routing protocols to achieve a network-wide lifetime maximization. In the next chapter, we present our proposition for an energy-efficient routing called $O(1)$ -reception routing. We also show how it can be used with the proposed MAC protocols and the lifetime extension we obtain by using them jointly.

7 Evaluating Link Cost and Reliability of Frame Preamble Protocols

8 $O(1)$ -Reception Routing

The methods proposed in the previous chapters (4 through 7) reduce the energy consumption at the MAC layer, that is, they help each single node to manage its own resources independently of the other nodes to increase its lifetime. Although these methods contribute to increasing the lifetime of the whole network by increasing the lifetime of each single node, they are insufficient alone. The lifetime of a sensor network also depends on the role of nodes: an overused node that prematurely dies may cause the failure of the whole network. Therefore, maximizing the lifetime of a sensor network also requires an energy-efficient routing protocol on top of an energy-efficient MAC protocol.

In Chapter 3, we have shown that an energy efficient routing protocol should minimize its complexity and its overhead while selecting energy efficient routes. Reducing the overhead is usually achieved by reducing the number of exchanged routing messages. However, selecting energy-efficient routes requires the use of a combined metric that takes into account the link cost (referred to as the min metric) to minimize the per transmitted packet energy, and the residual energy (referred to as the max-min metric) to avoid overusing vulnerable nodes.

Traditional energy-efficient routing protocols require the exchange of routing messages with energy information measured according to a considered metric: to select the best route, a node needs to receive routing messages from all of its neighbors in order to compare them and select the best downstream neighbor to which it forwards data. However, as a node eventually selects only one downstream neighbor, we argue that neither the reception nor the processing (comparison) of all the messages are needed. Thereby, we propose a protocol, called $O(1)$ -reception routing, that enables the best route selection based on exactly one message reception.

The key idea of the $O(1)$ -reception routing is based on delaying forwarding of routing messages for a time interval that is inversely proportional to the residual energy of nodes. This intentionally added delay influences the propagation of the routing messages so that the message coming from the best downstream neighbor is received the first. Thus, all the subsequent routing messages coming from the other neighbors can be ignored.

The $O(1)$ -reception routing has the following major advantages. First, it selects energy-efficient routes according to a combined min and max-min metric while reducing the amount of exchanged routing messages. Second, it can be used with a data-centric routing such as directed diffusion, which allows more energy savings through data aggregation.

The remainder of this chapter is organized as follows. We first present motivations and the principles of the $O(1)$ -reception routing scheme (Section 8.1). Next, we formulate the problem of choosing an adequate mapping function that turns residual energy into intentional delay (Section 8.2). Then, we consider two approaches, one based on heuristic functions (Section 8.3) and another one based on providing an exact solution (Section 8.4). We evaluate the proposed solutions analytically (Section 8.5) and through simulation (Section 8.6). Finally, we conclude (Section 8.7).

8.1 Overview of $O(1)$ -Reception Routing

The $O(1)$ -reception routing is based on our energy-delay mapping technique. Therefore, it can be used to enhance any min-delay routing scheme including Directed Diffusion (see Section 3.2.1). Directed Diffusion is destination-initiated in the sense that data collectors (also called sinks) query data publishers (also called sources) asking for specific data types. This phase, similar to a route request in on-demand ad hoc routing protocols, is called interest propagation. It establishes localized data-forwarding pointers (called gradients) from sources to sinks. Sources then stream the requested data back to sinks according to the directions indicated by the gradients. Although there are different implementations of gradient routing, one phase pull directed diffusion is the best fit when few sinks collect the data published by many sources. Since such situations are fairly common in sensor network applications, we only consider one phase pull directed diffusion¹ in this chapter.

Our motivations for using diffusion are the following:

Computational complexity is reduced to a minimum. Each node only needs to broadcast one interest message during the interest propagation phase and it only needs to receive one interest message to setup its routing table (it can ignore the subsequent interest messages related to that same interest). Note that this situation is beneficial only if the underlying MAC protocol enables filtering redundant messages. In the previous chapters, we have shown how to enable this feature in both preamble sampling and common active/sleep schedules based MAC protocols. Filtering redundant messages allows a node to switch its radio off during redundant receptions, which saves energy.

There is no overhead due to the exchange of extra information like hello or route metrics messages, which saves more energy and reduces the computation complexity and memory occupation of the routing protocol.

Routing tables only require one entry per active interest consisting of a pointer toward the next node downstream.

It enables in-network processing to aggregate data based on attributes used in diffusion, which saves more energy by reducing the amount of transmitted and received messages.

The $O(1)$ -Reception Routing enhances the basic diffusion routing scheme by delaying the interests forwarding for an interval inversely proportional to the residual energy: nodes compute a forwarding delay based on their residual energy and defer the forwarding of interest messages for this period of time. As maximum lifetime routing should combine the min and the max-min metrics, the energy-delay mapping function should have the following properties: nodes with high residual-energy forward interests without delay to make diffusion equivalent to the min energy routing, and nodes with low residual-energy delay forwarding of interests for a time interval to make diffusion equivalent to the max-min residual energy routing.

¹which we will simply call diffusion.

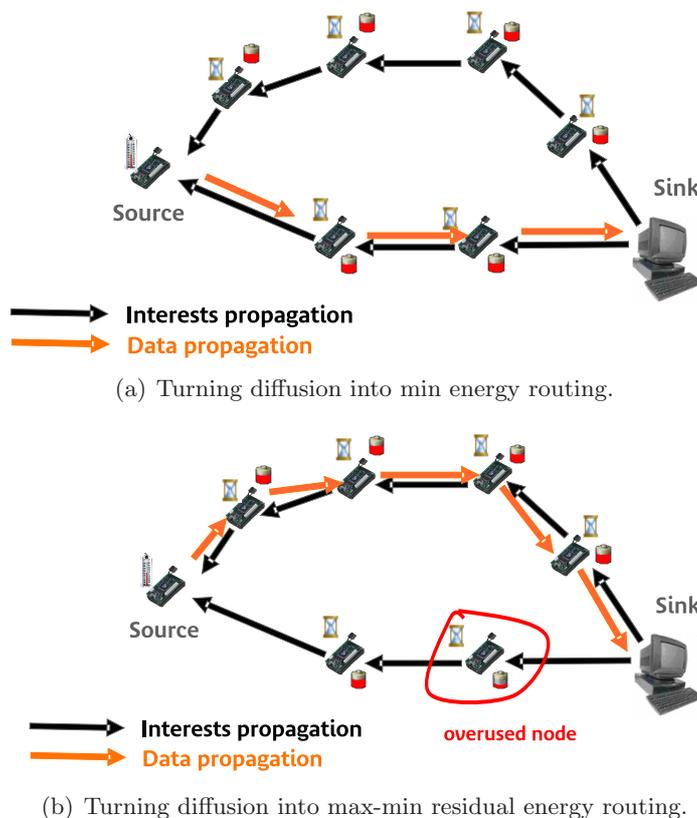


Figure 8.1: Principles of energy-delay mapping technique.

In Figure 8.1, we present the principles of such an energy-delay mapping technique. In Figure 8.1(a), all nodes have high residual energies thus they do not add any intentional delay. Therefore, the selected route is the shortest one as the interests propagation on which is the fastest. In our case, this corresponds to the min energy routing as the shortest route is the minimum energy consumption route, because we consider that all links have the same energy consumption. In Figure 8.1(b), we illustrate the max-min part of the algorithm that is used to protect nodes with low residual energies. The overused node shown in Figure 8.1(b), which is on the shortest route, should choose its intentional forwarding delay so that interest propagation on the other route is faster and thus the route is selected according to the max-min metric.

8.2 Problem Statement and System Model

The main problem in our routing scheme is energy-delay mapping, i.e. how to relate the residual energy to the intentional delay. Turning a min-delay metric into the min-energy metric is fairly straightforward when all links have equal energy consumption: it is sufficient to add no extra intentional delay. However, turning a min-delay metric into the max-min residual-energy metric is much more complex. Therefore, we formulate the problem as follows.

8 $O(1)$ -Reception Routing

Assume \mathcal{R} is the set of all possible routes between a sink node and a source node. We call $|R_k|$ the number of intermediate nodes on route R_k ($R_k \in \mathcal{R}$), source and destination nodes are not included. We use the following notation to represent R_k , $R_k = N_{1k} - \dots - N_{i_k} - \dots - N_{|R_k|k}$, where N_{i_k} represents an intermediate node on route R_k .

We assume that each node is able to measure its residual energy and we call ζ_{i_k} the relative residual energy of node N_{i_k} . Values ζ_{i_k} are normalized in $[0, 1]$, i.e. $0 \leq \zeta_{i_k} \leq 1$ for all nodes.

We call ζ_k^- the node with the least amount of residual energy on route R_k . We have:

$$\zeta_k^- = \min_{1 \leq i \leq |R_k|} \left\{ \zeta_{i_k} \right\} \quad (8.1)$$

The max-min residual energy routing selects the route with the largest ζ_k^- , i.e. it selects the route R that satisfies:

$$R = \operatorname{argmax}_{R_k \in \mathcal{R}} \left\{ \zeta_k^- \right\} \quad (8.2)$$

By combining Eq. 8.1 and Eq. 8.2, we obtain:

$$R = \operatorname{argmax}_{R_k \in \mathcal{R}} \left\{ \min_{1 \leq i \leq |R_k|} \left\{ \zeta_{i_k} \right\} \right\} \quad (8.3)$$

Let us now examine min-delay routing. We call D_{i_k} the delay introduced by each node N_{i_k} on route R_k . Route R_k experiences the total delay of $D(R_k)$. We have:

$$D(R_k) = \sum_{i=1}^{|R_k|} D_{i_k} \quad (8.4)$$

The min-delay routing selects the route with minimum delay. Therefore, the selected route, denoted by R' , satisfies:

$$R' = \operatorname{argmin}_{R_k \in \mathcal{R}} \left\{ D(R_k) \right\} \quad (8.5)$$

By combining Eq. 8.4 and Eq. 8.5, we obtain:

$$R' = \operatorname{argmin}_{R_k \in \mathcal{R}} \left\{ \sum_{i=1}^{|R_k|} D_{i_k} \right\} \quad (8.6)$$

Our goal is to make the min-delay routing select the route that satisfies the max-min residual energy metric, i.e. make route R' match route R . The next sections show how we achieve this goal by two different means: heuristics and exact solutions.

8.3 Approximate Solution: Heuristic Functions

To make route R' match route R , we propose to use a function f to map the residual energy of nodes into an intentional delay. Our goal is to solve Eq. 8.3 by solving Eq. 8.6 on a suitable set of:

$$D_{ik} = f(\zeta_{ik}) \quad (8.7)$$

By choosing f to be strictly decreasing, we can rewrite Eq. 8.3 as:

$$R = \operatorname{argmin}_{R_k \in \mathcal{R}} \left\{ f \left(\min_{1 \leq i \leq |R_k|} \left\{ \zeta_{ik} \right\} \right) \right\} \quad (8.8)$$

By matching Eq. 8.8 with Eq. 8.6 and replacing D_{ik} by its values calculated in Eq. 8.7, we conclude that function f that meet our goal should satisfy the following equation (Eq. 8.9) for all i in $1, \dots, |R_k|$:

$$\sum_{i=1}^{|R_k|} f(\zeta_{ik}) = f \left(\min_{1 \leq i \leq |R_k|} \left\{ \zeta_{ik} \right\} \right) \quad (8.9)$$

We can obtain an approximate solution by choosing f to be convex and decreasing in $[0, 1] \rightarrow [0, 1]$ so that the minimal ζ_k^- along route R_k makes a dominant contribution to the sum to the left of Eq. 8.9, i.e. we have

$$f(\zeta_k^-) \gg \left(\sum_{i=1}^{|R_k|} f(\zeta_{ik}) - f(\zeta_k^-) \right). \quad (8.10)$$

therefore

$$\sum_{i=1}^{|R_k|} f(\zeta_{ik}) \approx f \left(\min_{1 \leq i \leq |R_k|} \left\{ \zeta_{ik} \right\} \right). \quad (8.11)$$

which is an approximate solution for Eq. 8.9

To find a mapping function f with suitable properties, we have explored a family of decreasing convex functions of the form $(1/x)^\eta$, where η is a positive parameter. We have shifted and shrunk them so that they map the residual energy in $[0, 1]$ into the normalized delay in $[0, 1]$. In Figure 8.2, we present the resulting set of functions labeled f_η with η taking integer values from 1 to 4.

We use the parameter η to control the convexity of the mapping function that determines its ability to approximate max-min routing. The purpose of the parameter is to influence the intentional delay applied by the node with the minimum residual energy on a route so that it will be dominant. In this way, the route with the max-min residual energy will be selected, because the interest propagation on this route will have minimal delay. The convexity determines the precision of the approximation in Eq. 8.11: the more convex the mapping

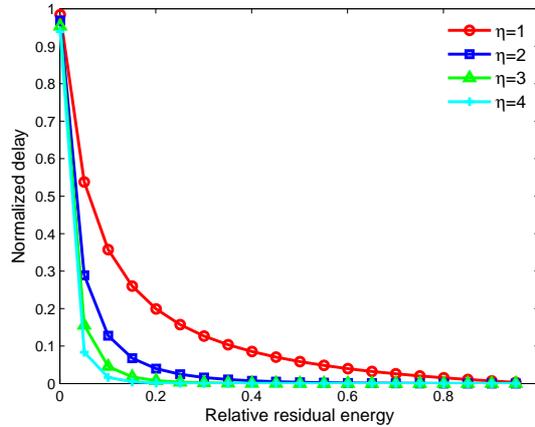


Figure 8.2: *Heuristic Mapping Functions.*

function, the better the approximation. For example, function f_4 has stronger convexity than the other functions in the considered set. Therefore, it approximates max-min routing better. However, functions with very high convexity such as f_4 present an inherent drawback resulting from their weak sensitivity threshold. The sensitivity threshold is the value that separates the flat part of the function from the curvy one. For example, function f_3 has a sensitivity threshold of 0.5, which means that a node using this mapping function will not apply any intentional delay when its residual energy is larger than 0.5. Therefore, if we have routes with nodes having residual energies larger than 0.5, the selected route will be the one with the min-delay, which very likely corresponds to the shortest path consuming the minimum energy. We can say that function f_3 uses a maximum lifetime routing with a battery protection threshold of 0.5. The battery protection threshold differentiates low residual-energy nodes from high residual-energy ones.

Note that there is a relation between the convexity of the function and the sensitivity threshold. More accurate max-min routing requires higher convexity functions (e.g. f_4), which results in smaller battery protection thresholds as higher convexity functions have smaller sensitivity thresholds. Vice versa, larger battery protection thresholds imply using larger sensitivity-threshold functions (e.g. f_1), which results in less max-min precision.

To overcome this shortcoming, we propose in the next section a synthetic mapping function that performs an exact transformation of the min metric into the max-min one according to an uncorrelated predefined battery protection threshold. This mapping function is to be used in the situation in which the residual energies of nodes are expressed as step functions and not continuous ones.

8.4 Exact Solution: Synthetic Function

In real implementations of routing protocols, the energy-delay mapping function would likely be discrete and tabulated. A node may read its battery voltage or integrate the consumed current and perform a table lookup to get the corresponding level of its residual energy. Therefore, we can assume that the residual energy of nodes is discrete.

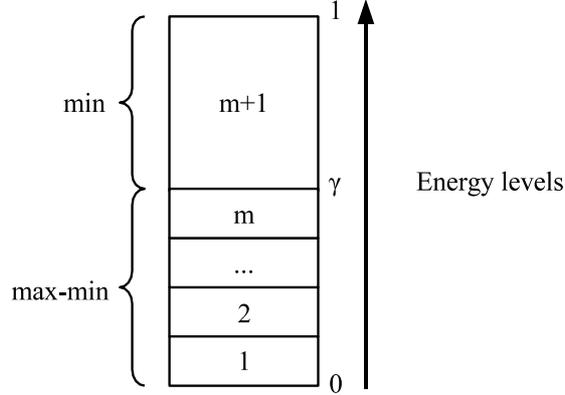


Figure 8.3: Energy Levels.

We call γ the battery protection threshold, ($0 < \gamma < 1$). Then, a node is *vulnerable*, if its residual energy is less than battery protection threshold γ . A node is *critical* for a route, if it has the least amount of residual energy among all the nodes forming that route. The *residual energy of a route* is equal to the residual energy of the critical node for that route. A *route is vulnerable*, if its residual energy is less than γ .

We aggregate all the energy levels greater than γ into one energy level as shown in Figure 8.3. We quantize the energy level below γ in m even levels: a discrete energy level l corresponds to the residual energy ζ if:

$$(l-1)\frac{\gamma}{m} < \zeta \leq l\frac{\gamma}{m}. \quad (8.12)$$

If ζ is larger than γ , the node is assigned discrete energy level $m+1$. Therefore, we have:

$$l = \begin{cases} \left\lceil \frac{m\zeta}{\gamma} \right\rceil & \text{if } \zeta \leq \gamma \\ m+1 & \text{otherwise.} \end{cases} \quad (8.13)$$

Let g be a synthetic function that maps residual energy into intentional forwarding delay d : $d = g(\zeta)$. As we use discrete energy levels instead of continuous residual energy, function g is dependent on m . Therefore, the intentional forwarding delay $d(l)$ that corresponds to energy level l is the following:

$$d(l) = g_m(l). \quad (8.14)$$

The synthetic function g that meets our goals even in the worst case needs to be decreasing so that $g(l) < g(l')$ for all $l > l'$. In addition, g also needs to be convex to mitigate the effect of increasing delay cumulated along longer routes. In Figure 8.4, we shows the worst case with an example with two routes R_k and $R_{k'}$. Route R_k has the maximum route length $|R_k| = n$ and a residual energy level of l . However, route $R_{k'}$ has the minimum route length $|R_{k'}| = 1$ and a residual energy level of $l-1$. To select the best route (route R_k), the interest propagation delay $D(R_k)$ on route R_k should be less than $D(R_{k'})$. As we assume these delays to be discrete, it is sufficient to have:

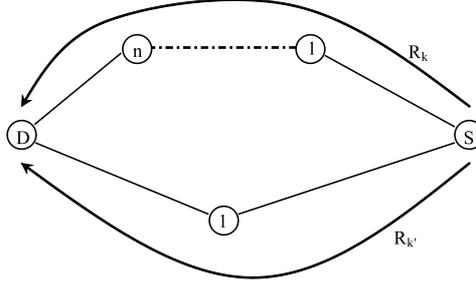


Figure 8.4: *The worst case illustrated with two routes.*

$$D(R_{k'}) = D(R_k) + 1. \quad (8.15)$$

therefore,

$$\sum_{i=1}^{|R_{k'}|} D_{ik'} = \sum_{i=1}^{|R_k|} D_{ik} + 1, \quad (8.16)$$

where D_{ik} is the delay incurred by node N_{ik} . Actually, the delay D_{ik} is composed of two parts: an intentional delay d_{ik} controlled via the synthetic mapping function and an inherent system delay δ_{ik} that includes computation and transmission delays. For example, in contention-based medium access protocols, the system delay includes the average backoff time used to reduce collision rates. Thus, we have

$$D_{ik} = d_{ik} + \delta_{ik}. \quad (8.17)$$

In the worst case, nodes on route R_k experience maximum system delays, i.e. $\delta_{ik} = \delta_{max}$ and nodes on route $R_{k'}$ experience minimum system delay $\delta_{ik'} = 0$. Moreover, all nodes on route R_k have their energy levels equal to l , i.e. $d_{ik} = g_m(l)$ for $i = 1, \dots, |R_k|$ and the node on route $R_{k'}$ has its energy level equal to $l - 1$, i.e. $d_{ik'} = g_m(l - 1)$ for $i = 1, \dots, |R_{k'}|$. Therefore, Eq. 8.16 can be rewritten as:

$$g_m(l - 1) = n [g_m(l) + \delta_{max}] + 1. \quad (8.18)$$

We set $g_m(m + 1)$ to 0 so that non vulnerable nodes do not apply any intentional delay, which performs min energy routing without any added delay. Therefore, Eq. 8.18 rewrites to as:

$$g_m(l) = \begin{cases} (n\delta_{max} + 1) \frac{n^{m-l+1} - 1}{n - 1} & \text{if } l \leq m \\ 0 & \text{otherwise.} \end{cases} \quad (8.19)$$

Table 8.1: *Notation*

p_γ	probability that a node is not vulnerable
$ \mathcal{R} $	number of disjoint routes between the source and the sink
$ R_k $	length of route R_k
n	number of intermediate nodes on the longest route between the source and the destination
$P_{min}(k)$	probability that route R_k is not vulnerable
$P_{maxmin}(k)$	probability that route R_k is vulnerable
P_{maxmin}	probability that the node selects a vulnerable route

8.5 Analytical Evaluation

As the synthetic function is more suitable for real implementations than the heuristic and theoretical function, we evaluate in this section the end-to-end intentional added delay for these functions. We also analyze the rate of vulnerable routes as the intentional delay applies only to vulnerable routes.

8.5.1 Worst Case Interest Propagation Delay

Assume that there are n intermediate nodes N_1, \dots, N_n between the source and the destination. Each node N_i has residual energy level l_i . On route $R = N_1 - \dots - N_n$, node N_i receives the interest at time t_i (we assume the destination sends the interest at time 0):

$$\left\{ \begin{array}{l} t_1 = \delta_1 \\ t_2 = (g(l_1) + \delta_2) + \delta_1 \\ t_3 = (g(l_2) + \delta_3) + (g(l_1) + \delta_2) + \delta_1 \\ \vdots \\ t_{n+1} = \sum_{i=1}^n (g(l_i) + \delta_{i+1}) + \delta_1 \end{array} \right. \quad (8.20)$$

where t_{n+1} is the time when the source receives the interest.

In the worst case, all intermediate nodes N_i , $i = 1, \dots, n$ have residual energy levels of 1 (i.e. $l_i = 1$ for all $i = 1, \dots, n$) and all system delays $\delta_i = \delta_{max}$ for all $i = 1, \dots, n$. Hence, the maximum interest propagation delay in the worst case corresponds to the maximum value of t_{n+1} , which is:

$$\begin{aligned} D_{max} &= n(n^{m-1} - 1) \left(\delta_{max} + \frac{1}{n-1} \right) \\ &= O(n^m \delta_{max}). \end{aligned} \quad (8.21)$$

8.5.2 Vulnerable Routes Rate

We propose to analyze the probability with which a node uses min or max-min metrics to select routes. This probability depends on parameters shown in Table 8.1.

8 $O(1)$ -Reception Routing

A node picks out a route according to the max-min metric if all the routes are vulnerable. Then,

$$\begin{aligned} P_{maxmin} &= \prod_{k=1}^{|\mathcal{R}|} P_{maxmin}(R_k) \\ &= \prod_{k=1}^{|\mathcal{R}|} \left(1 - P_{min}(R_k)\right) \end{aligned} \quad (8.22)$$

A route is not vulnerable if all the intermediate nodes on that route are not vulnerable. Therefore,

$$P_{min}(R_k) = \prod_{i=1}^{|R_k|} p_\gamma, \quad (8.23)$$

and,

$$P_{maxmin} = \prod_{k=1}^{|\mathcal{R}|} \left(1 - \prod_{i=1}^{|R_k|} p_\gamma\right) \quad (8.24)$$

The mean $E[P_{maxmin}]$ is the following:

$$E[P_{maxmin}] = \left(E[1 - p_\gamma^L]\right)^{|\mathcal{R}|}, \quad (8.25)$$

where L is a random variable that expresses route lengths. We have

$$E[1 - p_\gamma^L] = \sum_{i=1}^n (1 - p_\gamma^i) \cdot P\{L = i\} \quad (8.26)$$

We assume L being a discrete uniform random variable in $[1, n]$, i.e. $P\{L = i\} = 1/n$. Thus

$$E[1 - p_\gamma^L] = \frac{1}{n} \left(n - \sum_{i=1}^n p_\gamma^i\right). \quad (8.27)$$

Finally,

$$E[P_{maxmin}] = \left[1 - \frac{p_\gamma}{n} \left(\frac{1 - p_\gamma^n}{1 - p_\gamma}\right)\right]^{|\mathcal{R}|} \quad (8.28)$$

From Eq. 8.28 and Figure 8.5, we conclude that the probability of selecting a route according to max-min (i.e. all the routes are vulnerable) decreases when the number of routes $|\mathcal{R}|$ increases. This means that in dense networks in which there are many alternative routes, finding a not vulnerable route becomes very likely. We also notice that probability P_{maxmin} increases when the number of intermediate nodes n increases, which is quite expected. Besides, when probability p_γ that a node is not vulnerable increases, probability P_{maxmin} that all the routes are vulnerable decreases, because the number of vulnerable nodes decreases.

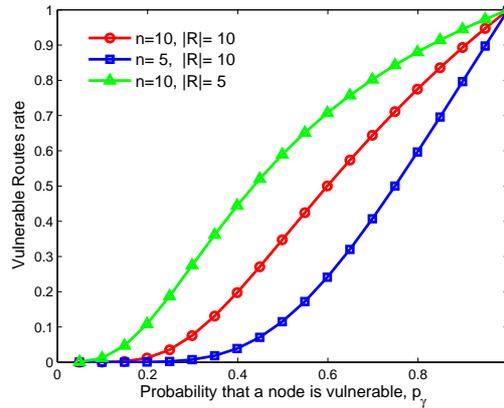


Figure 8.5: The rate of vulnerable routes in function of the probability that a node is vulnerable. $|\mathcal{R}|$ is the number of disjoint routes between the source and the destination.

8.6 Simulations

We have used ns-2 [89] to evaluate our synthetic mapping functions when used with diffusion. The goal of these experiments is to observe the lifetime extension obtained through the use of our mapping functions, the corresponding end-to-end interest propagation delay, and the benefits of filtering redundant interests. As the lifetime extension also depends on the energy-efficiency of the MAC protocol beneath, we have run simulations with two types of MAC protocols: an ideal MAC and MFP (see Chapter 6). The ideal MAC has no idle listening (i.e. a node consumes energy only when it transmits or receives a message) and no collisions. Therefore, it allows us to quantify the benefit of the mapping function independently of the MAC protocol performance. The MFP protocol allows us to show the expected performance with an real MAC protocol. To evaluate the performance of filtering redundant interests, we use an ideal MAC that filters out redundant messages before their receptions. We call this protocol Ideal-filter to distinguish it from Ideal-nofilter that does not filter out redundant messages. Likewise, we activate the filtering option of MFP in MFP-filter and we distinguish it from MFP-nofilter.

We have carried out experiments on two networks topologies: a random topology (Figure 8.6) and a star topology (Figure 8.7). In the former, the sink is Node 0 and the sources are Nodes 1 through 5. In the latter, the sink is Node 0 and the sources are Nodes 1 through 8. The sink generates interests every 100 seconds for refreshing existing routes or finding new ones. Every source that receives an interest sends data back to the sink according to the gradient installed by the interests. Each source sends a data message every 30 seconds. We have used a simple energy model in which transmission and reception powers are equal. The sink have unlimited initial energy, whereas the other nodes including sources have enough initial energy so that a significant amount of data messages are received by the sink from each source.

For the experiments, we consider two performance parameters: the lifetime extension achieved by the $O(1)$ -reception routing protocol compared to the basic diffusion protocol for each source and the end-to-end interest propagation delay from the sink to sources. We

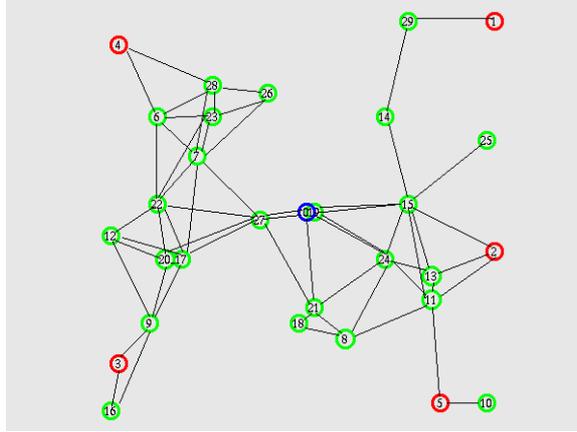


Figure 8.6: *Random Network.*

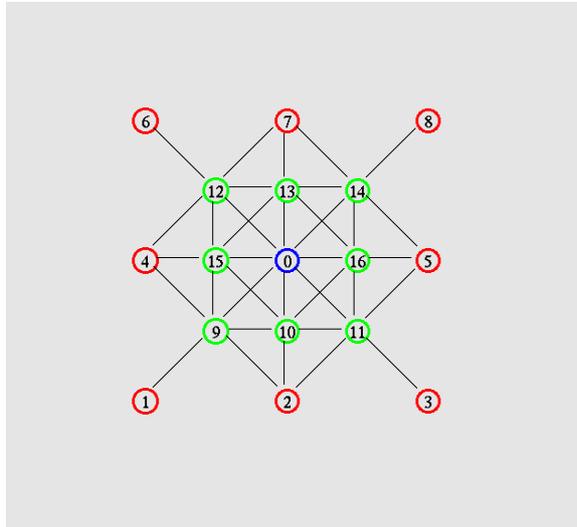


Figure 8.7: *Star Network.*

calculate the lifetime of each source in function of the number of transmitted data messages that successfully reach the sink before the source loses connection with the sink.

In the first experiments, we have set the number of energy levels m to 4 and varied the battery protection threshold γ from 0.1 to 0.9. We have measured the resulting lifetime extensions obtained in the best case for both topologies, i.e. when an Ideal-filter MAC protocol is used. In Figure 8.8, we plot two measures of the lifetime extension: the avg-lifetime and the max-lifetime. The avg-lifetime is obtained by averaging out all the lifetime extensions by all the sources, and the max-lifetime is the lifetime extension of the source that obtained the maximum lifetime extension. In Figure 8.8, we show that the random and the star topologies have the same behavior: the avg-lifetime and the max-lifetime increase when the battery protection threshold increases for both topologies. Therefore, we conclude that a large battery threshold is better for these situations. For the next experiments, we

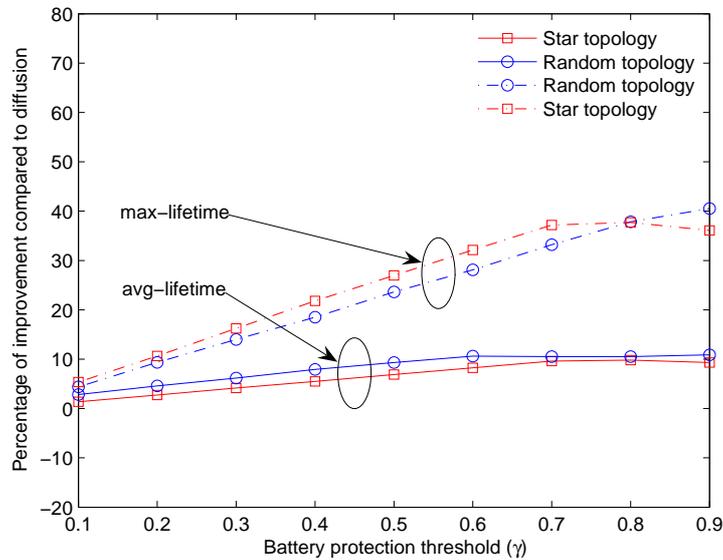


Figure 8.8: Lifetime extension according to γ with the Ideal-filter MAC protocol. The value of m is set to 4.

set the battery protection threshold γ to 1.

For the second experiments, we have varied m from 1 to 9 to evaluate the tradeoff between the lifetime extensions and the end-to-end interest propagation delays. In Figure 8.9, we show the lifetime extensions obtained with Source 1 (Figure 8.9(a)) and Source 2 (Figure 8.9(b)) in the star topology. We have plotted lifetime extensions only for these sources because there is a symmetry in the star topology: the results obtained for Sources 3, 6, and 8 are the same as those obtained for Source 1, and those obtained for nodes 4, 5, and 7 are the same as those obtained for Source 2.

As expected, Figure 8.9(a) shows that the lifetime extension increases when the number of levels m increases, because the more energy levels we have, the more accurate our mapping function is. However, the percentage of lifetime extension increases with less intensity. That is, increasing m from 2 to 3 increases the lifetime extension by a factor that is smaller than that when increasing m from 1 to 2.

Note that increasing the lifetime of some sources may decrease that of other sources, which results in some sources with negative lifetime extensions as shown in Figure 8.9(b). Source 2 (and also Sources 4, 5, and 7) has a negative lifetime extension, because their lifetime with diffusion routing is longer than that with $O(1)$ -Reception Routing. With diffusion, Source 2 has three potential relays (Nodes 9, 10, and 11). However, with $O(1)$ -Reception Routing, Source 2 mostly has only one relay (Node 10), because Nodes 9 and 11 relay the traffic of Sources 1 and 3, respectively. Note that even with these negative lifetime extensions, the overall lifetime extension (the avg-lifetime in Figure 8.8) is positive.

We have carried out the same experiments on the random topology and obtained the following results: Sources 1 and 5 have positive lifetime extensions, Sources 3 and 4 have zero lifetime extensions, and Source 2 has negative lifetime extension. Source 3 and 4 have

8 $O(1)$ -Reception Routing

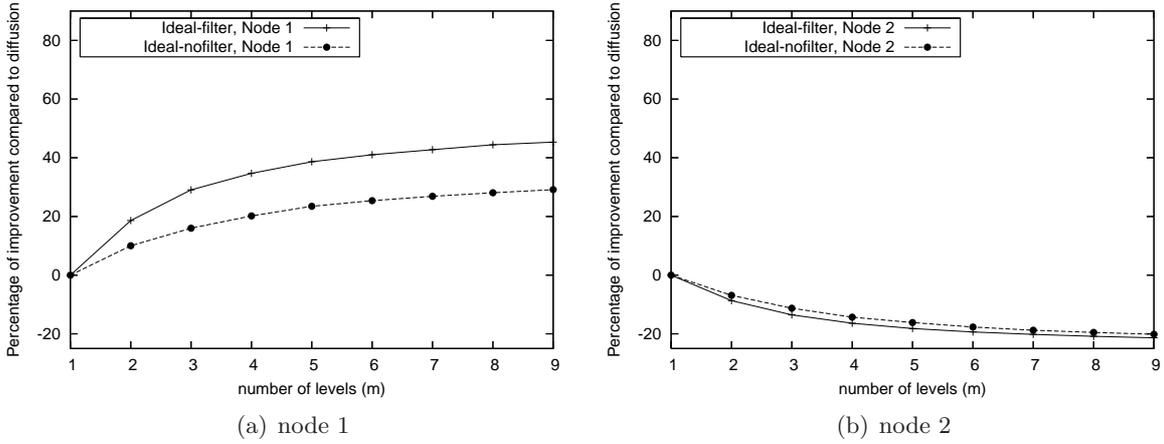


Figure 8.9: Percentage of improvement with $\gamma = 1.0$ for a star network.

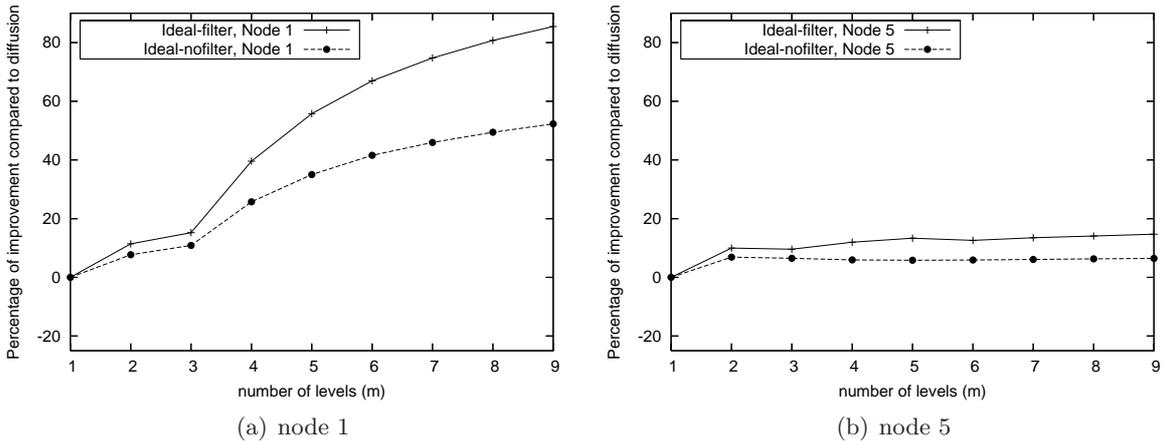


Figure 8.10: Percentage of improvement with $\gamma = 1.0$ for a random network.

zero lifetime extensions, because all their traffic passes through Node 27. As there are no alternative routes for these sources, no lifetime extension will be achieved no matter how well the routing algorithm performs. Source 2 has a negative lifetime for the same reasons explained above with the star topology. For Sources 1 and 5, Figure 8.10 shows the percentage of their lifetime extensions. We can see that the lifetime extension for Source 1 is larger than that for Source 5. Two key nodes (15 and 11) that are critical for the lifetime of Sources 1 and 5 respectively cause this result. With diffusion, Source 1 has a lifetime that is smaller than that of Source 5, because the only route that connects Source 1 with the sink contains Node 15. This route is more vulnerable than the other routes connecting Source 5 to the sink, because Node 15 relays most of the traffic of Source 2 as it is on the shortest route from Source 2 to the sink, which is route $(2 - 15 - 0)$. Moreover, Node 15 is more vulnerable than Node 11, because it receives and sends more interests as it has a higher number of neighbors. As our algorithm protects vulnerable nodes from being overused, the

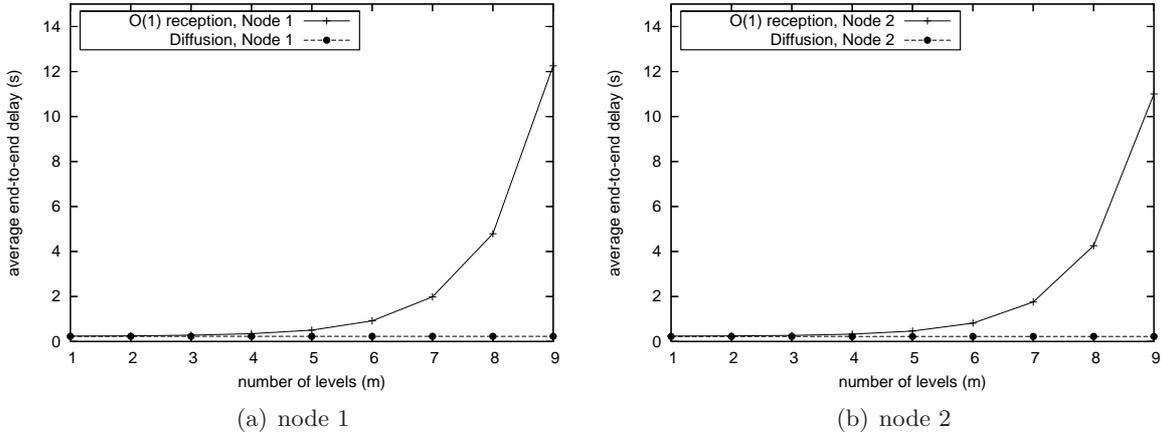


Figure 8.11: Average interest propagation delay with $\gamma = 1.0$ for a star network.

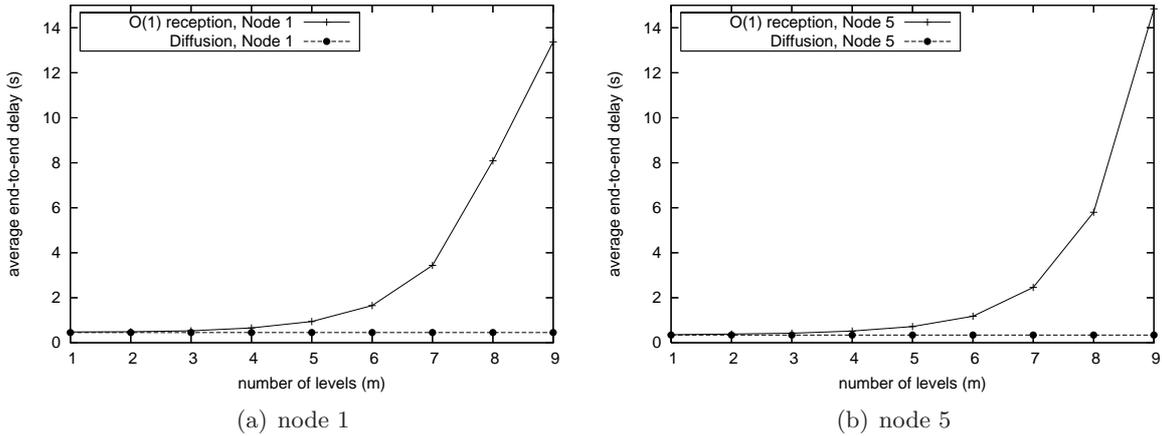


Figure 8.12: Average interest propagation delay with $\gamma = 1.0$ for a random network.

lifetime of Node 15 increases with a percentage that is larger than that of Node 11, thus increasing the lifetime of Sources 1 and 5 accordingly.

From Figure 8.9(a), Figure 8.10(a), and Figure 8.10(b), we can see that increasing the number of energy levels m increases the lifetime of sources connected to the sink through vulnerable routes. However, it is expected to increase the end-to-end interest propagation delays. Therefore, we need to make a trade-off between lifetime extension and interest propagation delays by choosing a suitable value for m . For this, we present in Figure 8.11 and Figure 8.12 the average end-to-end interest propagation delays experienced in diffusion and in $O(1)$ -reception routing for the star and the random topologies, respectively. These figures confirm the derivations carried out in Section 8.5.1 that show that the end-to-end interest propagation delay increases exponentially when m increases linearly. For example, when $m = 4$, we obtain a substantial lifetime extension with an almost negligible end-to-end interest propagation delay.

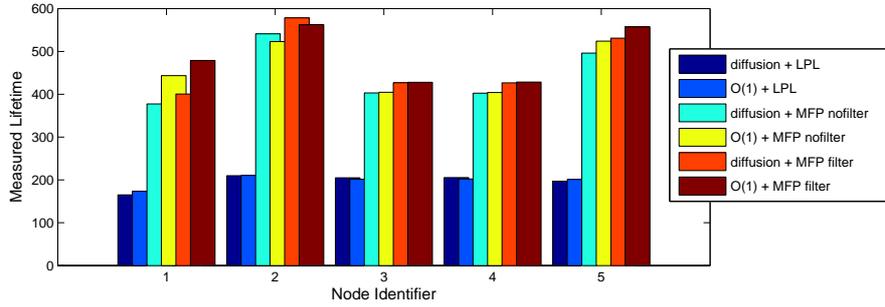


Figure 8.13: Lifetime extension for each node in the random topology.

In addition to increasing the lifetime of sources by using an energy-efficient metric, $O(1)$ -reception routing also increases the lifetime of the network by reducing the overhead of exchanged messages. Figure 8.10(a) shows that avoiding the reception of redundant messages at the MAC layer (Ideal-filter) allows Node 1 to increase its lifetime by up to 40% compared to when no filtering (Ideal-nofilter) is used. This significant lifetime extension percentage is mainly due to the improvements realized by Node 15 that has a large number of neighbors. When there is no filtering, Node 15 receives all the interests forwarded by its neighbors, i.e. 8 interests with the same information. However, when filtering is used, Node 15 receives only 1 interest as it filters out the redundant interests. We expect that filtering will achieve further energy saving in more dense networks.

In Fig 8.13, we plot the results we obtained for $O(1)$ -Reception Routing with realistic MAC protocols: LPL, MFP-filter, and MFP-nofilter. The results obtained for Sources 1 through 5 confirm the arguments presented above. They also show that we obtain a substantial² lifetime when jointly using our contributions: MFP-filter and $O(1)$ -Reception Routing.

8.7 Conclusion

Maximizing the lifetime of a sensor network requires an energy-efficient routing protocol on top of an energy-efficient MAC protocol. In this chapter, we have tackled the problem of selecting energy-efficient routes while reducing the overhead of routing protocols. We have proposed a technique called $O(1)$ -reception that enables the best route selection based on exactly one routing message reception, thus allowing substantial overhead reduction because in traditional routing a node needs to receive routing messages from all of its neighbors to be able to select the best route.

The $O(1)$ -reception routing is suitable for WSNs not only because it reduces reception overhead but also because it can be used with any metric that can be mapped on top of the min delay metric. In this chapter, we have proposed an example in which $O(1)$ -reception is used to perform a hybrid min and max-min routing with directed diffusion, allowing thus to benefit from the advantages of a data-centric communication scheme such as traffic

²Note that as explained in the previous section, the $O(1)$ -reception routing marginally reduces the lifetime of Node 2 when it aims at increasing the lifetime of Node 1 and Node 5.

aggregation.

The key idea of the $O(1)$ -reception routing is based on delaying the forwarding of routing messages for a time interval that is inversely proportional to the residual energy of nodes. This intentionally added delay influences the propagation of routing messages so that the first received one indicates the best route and thus all the subsequent routing messages with the same content can be ignored.

The intentional mapping delay is calculated according to a mapping function that determines the corresponding delay in function of the residual energy of a node. We have shown how to find such functions when the residual energy is continuous and when it is a discrete measure. As in practical implementations the residual energy is discrete, we have analytically evaluated the end-to-end intentional delay in the worst case and the percentage of vulnerable routes in which this delay is added.

We have run extensive simulations with ns-2 to evaluate the performance of $O(1)$ -reception routing. As the performance of energy efficient routing also depends on the MAC protocol beneath, we have considered two MAC protocols: an ideal MAC protocol to only evaluate the benefit of our routing protocol and a real MAC protocol (MFP) to evaluate their combined benefit. The obtained results show that using MFP jointly with $O(1)$ -reception routing achieves a substantial lifetime extension.

8 $O(1)$ -Reception Routing

9 Conclusions

Sensor networks promise many new exciting applications and services with potentially substantial benefits in various domains. However, many problems need to be solved before the wide deployment and proliferation of such networks. The major problem is energy management-as sensor nodes are very energy constrained so we need clever energy management mechanisms to maximize their lifetime and make sensor networks cost-effective.

Extending the lifetime of a sensor network requires considerable effort at both local and global scopes. Acting at such scopes is complementary: at the local scope, each node optimizes its energy consumption locally and independently of the whole network whereas at the global level, nodes cooperate to optimize the overall usage of energy resources. These two scopes require interdisciplinary optimizations covering many domains such as electronics, systems, and networking protocols as well as a cross-layer approach to protocol layer interactions.

In this dissertation, we have considered the problem of energy management in wireless sensor networks at the both scopes through optimizing communication protocols. Two major reasons have motivated this choice. First, the state of the art clearly shows that the main sources of energy consumption at the local scope stem from the inefficiency of MAC protocols to cope with various forms of energy waste such as idle listening, collisions, and overhearing. Second, cooperation between nodes at the global level is mostly affected by the performance of routing protocols. Our contributions not only provide more energy-efficient MAC and routing protocols optimized independently of each other, but also result in a cross-layer design in which MAC and routing protocols are jointly optimized.

At the MAC layer, we have dealt with two of the main sources of energy waste: collisions and overhearing. We have differentiated between two types of collisions: those caused by hidden nodes and those caused by visible nodes. We have modelled both of them and proposed solutions for reducing them. We have also identified two types of overhearing: the first one results from receiving irrelevant messages such as redundant messages during a flooding operation; the other one comes from receiving irrelevant signals such as the continuous preamble used in preamble sampling protocols. For irrelevant messages, we have proposed abstract frames that make it possible for a node to avoid receiving redundant messages. For irrelevant signals, we have changed the way the preamble is transmitted in sampling protocols so that a node does not need to keep to receiving the whole preamble. We have proposed a solution called MFP (Micro Frame Preamble) that avoids both types of overhearing. We have shown that MFP achieves substantial energy savings. We have generalized MFP to Frame Preamble MAC and studied the relation between the reliability and the energy cost of a wireless link with its four variants. For these variants, we have shown that reliability has a strong impact on the energy cost of communication in a wireless

channel subject to erroneous transmissions: it is beneficial to increase reliability by avoiding retransmissions because they consume a large amount of energy.

At the routing layer, we have considered two techniques that reduce the overhead of routing protocols and distribute the load of packet forwarding among nodes to avoid their premature energy exhaustion. We have shown that these two techniques can be combined into a single routing protocol—the $O(1)$ -reception routing protocol. This protocol reduces the overhead by limiting the receptions to only useful messages and balances the load of nodes by avoiding the selection of routes with vulnerable nodes.

The proposed MAC (MFP) and routing ($O(1)$ reception) techniques are energy efficient by themselves and are compatible with other routing and MAC protocols. However, using them together is even more beneficial as they exhibit features that have been designed for a joint usage. Specifically, the fact that $O(1)$ -reception routing limits receptions to only necessary messages is only effective in so far the MAC protocol beneath is able to identify the unnecessary messages to avoid their reception, which is the case of MFP.

Future Directions and Perspectives

A pertinent subject of future research is related to the Frame Preamble MAC protocol—the generalized version of our MFP MAC protocol. Our study of four variants of the Frame Preamble MAC protocol (Chapter 7) has shown that transmission errors have an important impact on the energy cost of a wireless links: according to the error rates, some Frame Preamble MAC variants are more energy-efficient than the others are. This observation suggests using an adaptive Frame Preamble MAC that switches from one variant to another according to the observed transmission error rate. Although this idea is promising as it continuously puts the Frame Preamble MAC in the state of its optimal performance, it also requires close cooperation between nodes to ensure a consistent operation of the protocol. For example, neighbors of a node should be aware of the protocol variant currently used so that they can communicate by means of the same variant. In addition to the consistence issue, mechanisms should be investigated to determine the best variant according to the experienced conditions. Our study, carried on a single link with a BSC channel, should be generalized to more channel models and should also take collisions into account.

The improvement of MAC protocols may also change the design of routing protocols. In many energy efficient MAC protocols, unicast communications provide more energy saving than broadcasts. This simple observation may have an influence on the design of routing protocols and applications—they may reduce the use of broadcast communications and promote unicasts or transform a broadcast into multiple unicast transmissions. In addition to these parameters, the adaptive Frame Preamble MAC may also contribute with other properties that may influence the design of routing protocols and applications. Therefore, routing protocols should follow the evolution of MAC protocols.

In parallel to these specific research directions, we have identified other generic trends that may contribute to the development of sensor networks. A first proposal consists in providing

more engineering tools for debugging, testing, simulating, and deploying sensor networks. These tools are very helpful for rapid prototyping and evaluating proposed solutions. A second proposal consists of careful definition of application needs and setting benchmarks accordingly. Such benchmarks will facilitate the comparison of proposed solutions under similar conditions.

9 *Conclusions*

Author Publications List

Published/Accepted Papers

1. A. Bachir, L. Samper, D. Barthel, M. Heusse, A. Duda. Link Cost and Reliability of Frame Preamble MAC Protocols, *In proceedings of IEEE IWWAN*, 2006.
2. T. Watteyne, A. Bachir, M. Dohler, D. Barthel et I. Aug-Blum. A low-energy adaptive cross-layer communication protocol for avoiding 1-hop neighborhood knowledge. *In proceedings of IEEE IWWAN*, 2006.
3. A. Bachir, D. Barthel, M. Heusse, A. Duda, O(1) Reception Routing for Sensor Networks, *Computer Communications*, accepted 2006.
4. A. Bachir, D. Barthel, M. Heusse, A. Duda, A Synthetic Function for Energy-Delay Mapping in Energy Efficient Routing, *In proceedings of IFIP WONS*, 2006.
5. A. Bachir, D. Barthel, M. Heusse, A. Duda, Micro-Frame Preamble MAC for Multihop Wireless Sensor Networks, *In proceedings of IEEE ICC*, 2006.
6. A. Bachir, D. Barthel, M. Heusse, A. Duda, Abstract Frames for Reducing Overhearing in Wireless Sensor Networks, *In proceedings of IFIP Networking*, 2006.
7. A. Bachir, D. Barthel, M. Heusse, A. Duda, Hidden Node Avoidance in Wireless Sensor Networks. *In proceedings of IEEE WirelessCom*, 2005.
8. A. Bachir, D. Barthel, Localized Maximum Minimum Energy Available Routing for WSN Using Delay Control. *In proceedings of IEEE ICC*, 2005.
9. A. Benslimane, A. Bachir, Ad hoc mobile networks: inter-vehicle geocast. *Book chapter: Management, Control and Evolution of IP Networks*. ISBN 1905209479. March 2006.
10. A. Bachir, A. Benslimane, Multicast dans les Réseaux Ad hoc, géodiffusion inter véhicules. *Chapitre du livre l'Internet Ambient*, pp 215-235, *Hermès science publications*, 2004.
11. A. Benslimane, A. Bachir, Inter-vehicle Geocast Protocol Supporting Non-equipped GPS Vehicles. *In Proceedings of ADHOC-NOW*, 2003.
12. A. Bachir, A. Benslimane, A Multicast Protocol in Ad Hoc Networks, Inter-Vehicle Geocast. *In proceedings of IEEE VTC*, 2003.

Submitted Papers

13. A. Bachir, S. Plancoulaine, D. Barthel, M. Heusse, A. Duda, Micro-Frame Preamble MAC for Multihop Wireless Sensor Networks, *IEEE Transactions on Mobile Computing*, November 2006.

Patents

1. D. Barthel, A. Bachir, *Procédé de routage, station émettrice-réceptrice et programme d'ordinateur pour la mise en oeuvre du procédé*. Filed by France Télécom R&D, 2005.
2. A. Bachir, D. Barthel, *Procédé et système de transmission d'un signal comprenant un préambule et une trame de données*. Filed by France Télécom R&D, 2005.
3. A. Bachir, L. Samper and D. Barthel, *Procédé de communication, stations émettrice et réceptrice et programmes d'ordinateur associés*, Filed by France Télécom R&D, 2006.
4. T. Watteyne, A. Bachir, M. Dohler and D. Barthel, Isabelle Augé-Blum. *Cross-layer communication protocol for avoiding 1-hop neighborhood knowledge*. Filed by France Télécom R&D, 2006.

Bibliography

- [1] T. He et al. Energy-Efficient Surveillance Systems Using Wireless Sensor Networks. *Proceeding of the ACM MobiSys*, Boston, MA, June 2004.
- [2] Christophe Dugas. Configuring and managing a large-scale monitoring network solving real world challenges for ultra-low powered and long-range wireless mesh networks. *International Journal of Network Management*, 15:269–282, 2005.
- [3] A. Mainwaring et al. Wireless Sensor Networks for Habitat Monitoring. *In Proceedings of ACM WSNA*, Atlanta, GA, September 2002.
- [4] R. Szewczyk et al. Habitat Monitoring with Sensor Networks. *Communications of the ACM*, 47(6):34–40, 2004.
- [5] E. Cayirci, T. Cuplu. SENDROM: Sensor Networks for Disaster Relief Operations Management. *In Proceedings of MedHocNet*, Bodrum, Turkey, June 2004.
- [6] J. Rabaey et al. PicoRadio Supports Ad Hoc Ultra-Low Power Wireless Networking. *IEEE Computer Magazine*, 33(7):42–48, 2000.
- [7] J. Hill et al. System Architecture Directions for Networked Sensors. *ACM SIGOPS Operating Systems Review*, 34(5):93–104, 2000.
- [8] J. Polastre, R. Szewczyk, D. Culler. Telos: Enabling Ultra-Low Power Wireless Research. *In Proceedings of IPSN/SPOTS*, pages 302–11, Los Angeles, CA, April 2005.
- [9] A. Bachir, S. Plancoulaine, D. Barthel, M. Heusse and A. Duda. Micro-Frame Preamble MAC for Multihop Wireless Sensor Networks. *Submitted to IEEE Transactions on Mobile Computing*, 2006.
- [10] Ember Corporation. EM250 Single-Chip ZigBee/802.15.4 Solution, Data Sheet. 2005.
- [11] Chipcon Corporation. CC2500 Single Chip Low Cost Low Power RF Transceiver, Data Sheet. 2005.
- [12] Freescale Semiconductor. MC13192/MC13193 2.4 GHz Low Power Transceiver for the IEEE 802.15.4 Standard. *Rev. 2.8*, 04/2005.
- [13] Z. Haas, J. Halpern, and L. Li. Gossip-Based Ad Hoc Routing. *In Proceedings of IEEE Infocom*, New York, NY, June 2002.
- [14] S.Y. Ni et al. The Broadcast Storm Problem in Mobile Ad Hoc Network. *In Proceedings of IEEE/ACM Mobicom*, Seattle, WA, August 1999.

Bibliography

- [15] M. Sun, W. Feng, and T.H. Lai. Location Aided Broadcast in Wireless Ad Hoc Networks. *In Proceedings of IEEE GLOBECOM 2001*, pages 2842–2846, San Antonio, TX, November 2001.
- [16] D. Braginsky and D. Estrin. Rumor Routing Algorithm for Sensor Networks. *In Proceedings of ACM WSNA*, Atlanta, GA, September 2002.
- [17] J. Cartigny, D. Simplot and I. Stojmenovic. Localized Minimum-Energy Broadcasting in Ad-hoc Networks. *In Proceedings of IEEE Infocom*, San Francisco, CA, April 2003.
- [18] V. Rodoplu and T. Meng. Minimum Energy Mobile Wireless Networks. *IEEE JSAC*, 8(17):1333–44, August 1999.
- [19] Toh, C.-K., Cobb, H. and Scott, D.A. Performance Evaluation of Battery-Life-Aware Routing Schemes for Wireless Ad Hoc Networks. *In Proceedings of IEEE ICC*, pages 2824–9, Helsinki, Finland, June 2001.
- [20] A. Misra and S. Banerjee. MRPC: Maximizing Network Lifetime for Reliable Routing in Wireless Environments. *In Proceedings of IEEE WCNC*, Orlando, FL, March 2002.
- [21] Qun Li and Javed A. Aslam and Daniela Rus. Online Power-Aware Routing in Wireless Ad Hoc Networks. *In Proceedings of ACM Mobicom*, pages 97–107, Rome, Italy, July 2001.
- [22] Q. Li and J. Aslam and D. Rus. Distributed Energy-Conserving Routing Protocols for Sensor Networks. *In Proceedings of IEEE HICSS*, Big Island, HI, January 2003.
- [23] R. C. Shah and J. M. Rabaey. Energy aware routing for low energy ad hoc sensor networks. *Proceeding of the IEEE WCNC*, Orlando, FL, March.
- [24] W. Ye, F. Silva and J. Heidemann. Ultra-Low Duty Cycle MAC with Scheduled Channel Polling. *In Proceedings of ACM SenSys*, Boulder, CO, November 2006.
- [25] C. Enz, A. El-Hoiydi, J. Decotignie, V. Peiris. WiseNET: An Ultralow-Power Wireless Sensor Network Solution. *IEEE Computer*, 37(8):62–70, August 2004.
- [26] M. Buettner et al. X-MAC: A Short Preamble MAC Protocol For Duty-Cycled Wireless Networks. *In Proceedings of ACM SenSys*, Boulder, CO, November 2006.
- [27] S. Mahlknecht and M. Boeck. CSMA-MPS: A Minimum Preamble Sampling MAC Protocol for Low Power Wireless Sensor Networks. *In Proceedings of IEEE Workshop on Factory Communication Systems*, Vienna, Austria, September 2004.
- [28] C. Schurgers et al. Optimizing Sensor Networks in the Energy-Latency-Density Design Space. *IEEE Transactions on Mobile Computing*, 1(1):70–80, January-March 2002.
- [29] A. Goldsmith, and S. Wicker. Design Challenges for Energy-Constrained Ad Hoc Wireless Networks. *IEEE Wireless Communications Magazine*, 9(4):8–27, August 2002.
- [30] Vikas Kawadia and P. R. Kumar. A Cautionary Perspective on Cross Layer Design. *IEEE Wireless Communications Magazine*, 12(1):3–11, February 2005.

- [31] R. Jurdak, C. V. Lopes, and P. Baldi. A Survey, Classification, and Comparative Analysis of Medium Access Control Protocols for Ad Hoc Networks. *IEEE Comm. Surveys and Tutorials*, 6(1):2–16, June 2004.
- [32] K. Langendoen and G. Halkes. Energy-Efficient Medium Access Control. *Book chapter in the Embedded Systems Handbook*, R. Zurawski (editor), CRC press, 2005.
- [33] I. Demirkol, C. Ersoy, and F. Alagoz. MAC Protocols for Wireless Sensor Networks: a Survey. *IEEE Communications Magazine*, 44(4):115–21, April 2006.
- [34] Y. Li, W. Ye, and J. Heidemann. Energy and Latency Control in Low Duty Cycle MAC Protocols. *In Proceedings of IEEE WCNC*, New Orleans, LA, March 2005.
- [35] Huan Pham and Sanjay Jha. An Adaptive Mobility-Aware MAC Protocol for Sensor Networks (MS-MAC). *In Proceedings of IEEE MASS*, pages 588–60, Fort Lauderdale, FL, October 2004.
- [36] M. Neugebauer, J. Plnngs and K. Kabitzsch. A New Beacon Order Adaptation Algorithm for IEEE 802.15.4 Networks. *In Proceedings of the EWSN*, pages 302–11, Istanbul, Turkey, January 2005.
- [37] H. Cao, K. Parker, and Anish Arora. O-MAC: A Receiver Centric Power Management Protocol. *In Proceedings of IEEE ICNP*, Santa Barbara, CA, November 2006.
- [38] IEEE 802.15.4. Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANS). 2003.
- [39] W. Ye, J. Heidemann, and D. Estrin. An Energy-Efficient MAC Protocol for Wireless Sensor Networks. *In Proceedings of IEEE Infocom*, pages 1567–76, New York, NY, July 2002.
- [40] A. Woo and D. Culler. A Transmission Control Scheme for Media Access in Sensor Networks. *In Proceedings of ACM Mobicom*, Rome, Italy, July 2001.
- [41] J. Polastre, J. Hill and D. Culler. Versatile Low Power Media Access for Wireless Sensor Networks. *In Proceedings of ACM SenSys*, 2004.
- [42] T. van Dam and K. Langendoen. An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks. *In Proceedings of ACM Sensys*, pages 171–80, Los Angeles, CA, November 2003.
- [43] IEEE 802.11. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. 1999.
- [44] W. Ye, J. Heidemann, D. Estrin. Medium Access Control with Coordinated, Adaptive Sleeping for Wireless Sensor Networks. *IEEE Trans. on Networking*, 12(3):493–506, June 2004.
- [45] P. Lin, C. Qiao and X. Wang. Medium Access Control with a Dynamic Duty Cycle for Sensor Networks. *In Proceedings of IEEE WCNC*, Atlanta, GA, March 2004.

Bibliography

- [46] G. Lu, B. Krishnamachari and C. Raghavendra. An Adaptive Energy-Efficient and Low-Latency MAC for Data Gathering in Sensor Networks. *In Proceedings of the WMAN*, Santa Fe, NM, April 2004.
- [47] A. Bachir, D. Barthel, M. Heusse and A. Duda. Abstract Frames for Reducing Overhearing in Wireless Sensor Networks. *In Proceedings of the IFIP Networking*, Coimbra, Portugal, May 2006.
- [48] A. El-Hoiydi. Aloha with Preamble Sampling for Sporadic Traffic in Ad Hoc Wireless Sensor Networks. *In Proceedings of IEEE ICC*, New York, NY, April 2002.
- [49] J. Hill and D. Culler. Mica: a wireless platform for deeply embedded systems. *IEEE Micro*, 22(6), 2002.
- [50] E-Y. Lin, J. Rabaey, A. Wolisz. Power-Efficient Rendez-vous Schemes for Dense Wireless Sensor Networks. *In Proceedings of IEEE ICC*, Paris, France, June 2004.
- [51] R. Jurdak, P. Baldi, and C. V. Lopes. Energy-Aware Adaptive Low Power Listening for Sensor Networks. *In Proceedings of INSS*, pages 24–29, San Diego, CA, June 2005.
- [52] I. Rhee et al. Z-MAC: a Hybrid MAC for Wireless Sensor Networks. *In Proceedings of ACM SenSys*, San Diego, CA, November 2005.
- [53] I. Rhee et al. DRAND: Distributed Randomized TDMA Scheduling For Wireless Adhoc Networks. *In Proceedings of ACM MobiHoc*, Florence, Italy, May 2006.
- [54] <http://www.zigbee.org> .
- [55] J. Polastre. *A Unifying Link Abstraction for Wireless Sensor Networks*. PhD thesis, University of California, Berkeley, October 2005.
- [56] S. Corson and J. Macker. Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations. *RFC 2501*, 1999.
- [57] E. Royer and C. Toh. A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks. *IEEE Personal Communications*, 6:46–55, December 1999.
- [58] H. Karl and A. Willig. *Protocols and Architectures for Wireless Sensor Networks*. Wiley edition, 2005.
- [59] J. N. Al-Karaki, A. E. Kamal. Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications*, 11(6):6–28, December 2004.
- [60] K. Akkaya and M. Younis. A Survey on Routing Protocols for Wireless Sensor Networks. *Ad Hoc Networks Journal*, 3(3):325–49, May 2005.
- [61] C. Intagonwiwat et al. Directed Diffusion for Wireless Sensor Networking. *IEEE/ACM Transaction on Networking*, 11(1):2–16, February 2003.
- [62] H. Takagi and L. Kleinrock. Optimal Transmission Ranges for Randomly Distributed Packet Radio Networks. *IEEE Transactions on Communications*, 32(3):246–57, 1984.

- [63] B. Karp and H. Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. *In Proceedings of ACM Mobicom*, Boston, MA, August 2000.
- [64] Y. Xu, J. Heidemann, and D. Estrin. Geography-informed Energy Conservation for Ad-hoc Routing. *In Proceedings of ACM Mobicom*, Rome, Italy, July 2001.
- [65] M. Zorzi and R. Rao. Geographic Random Forwarding (GeRaF) for Ad Hoc and Sensor Networks: Energy and Latency Performance. *IEEE Transactions on Mobile Computing*, 2(4):349–365, 2003.
- [66] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less Low Cost Outdoor Localization for Very Small Devices. *Technical Report 00729, Computer Science Dept.*, USC, April 2000.
- [67] A. Savvides, C. Han, and M. Srivastava. Dynamic Fine-Grained Localization in Ad Hoc Networks of Sensors. *In Proceedings of ACM Mobicom*, Rome, Italy, July 2001.
- [68] S. Capkun, M. Hamdi, and J. Hubaux. GPS-free Positioning in Mobile Ad Hoc Networks. *In Proceedings of IEEE HICSS*, Maui, HI, January 2001.
- [69] J. Heidemann et al. Building Efficient Wireless Sensor Networks with Low-Level Naming. *In Proceedings of ACM SOSP*, Lake Louis, Canada, October 2001.
- [70] F. Silva, J. Heidemann, and R. Govindan. Network Routing Application Programmer's Interface (API) and Walk Through 9.0.1. *Technical Report USC/ISI.*, December 2002.
- [71] J. Heidemann, F. Silva, and D. Estrin. Matching Data Dissemination Algorithms to Application Requirements. *In Proceedings of ACM Sensys*, pages 218–29, Los Angeles, CA, November 2003.
- [72] D. Ganesan et al. Highly Resilient, Energy Efficient Multipath Routing in Wireless Sensor Networks. *ACM Mobile Computing and Communications Review (MC2R)*, 1(2), 2002.
- [73] S. De, C. Qiao, and H. Wu. Meshed Multipath Routing: An Efficient Strategy in Sensor Networks. *In Proceedings of IEEE WCNC*, New Orleans, NA, March 2003.
- [74] Y. Yu, D. Estrin, and R. Govindan. Geographical and Energy Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks. *Technical Report 010023, Computer Science Dept.*, UCLA, May 2001.
- [75] R. Wattenhofer, L. Li, P. Bahl, and Y.-M. Wang. Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad Hoc Networks. *In Proceedings of IEEE Infocom*, pages 1388–97, Anchorage, AL, April 2001.
- [76] J. Gomez et al. Power-Aware Routing in Wireless Packet Radio. *In Proceedings of IEEE MoMuC*, San Diego, CA, November 1999.
- [77] J. Gomez et al. Conserving Transmission Power in Wireless Ad Hoc Networks. *In Proceedings of IEEE ICNP*, Riverside, CA, November 2001.

Bibliography

- [78] N. Bambos. Toward Power-Sensitive Network Architectures in Wireless Communications: Concepts, Issues, and Design Aspects. *IEEE Personal Communications*, 5(3):50–59, 1998.
- [79] J. Wu, F. Dai, M. Gao, I. Stojmenovic. On Calculating Power Aware Connected Dominating Sets for Efficient Routing in Ad Hoc Wireless Networks. *IEEE/KICS Journal of Communications and Networks*, 1(4):59–70, March 2002.
- [80] S. Singh, M. Woo, and C. Raghavendra. Power-Aware Routing in Mobile Ad Hoc Networks. In *Proceedings of ACM/IEEE Mobicom*, Dallas, TX, October 1998.
- [81] C. -K. Toh. Maximum Battery Life Routing to Support Ubiquitous Mobile Computing in Wireless Ad Hoc Networks. *IEEE Communications Magazine*, 39(6):138–47, June 2001.
- [82] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves. Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks. In *Proceedings of ACM Sensys*, pages 181–92, Los Angeles, CA, November 2003.
- [83] A. Bachir, D. Barthel, M. Heusse and A. Duda. Micro-Frame Preamble MAC for Multihop Wireless Sensor Networks. In *Proceedings of IEEE ICC*, Istanbul, Turkey, June 2006.
- [84] Tobagi, F. and L. Kleinrock. Packet Switching in Radio Channels: Part II—the Hidden Terminal Problem in Carrier Sense Multiple-Access and the Busy-Tone Solution. *IEEE Trans. On Comm.*, 23(12):1417–33, December 1975.
- [85] P. Karn. MACA- a new channel access method for packet radio. in *ARRL/CRRL Amateur Radio 9th Computer Networking*, pages 134–40, 1990.
- [86] Shugong Xu and Tarek Saadawi. Does the IEEE 802.11 MAC Protocol Work Well in Multihop Ad Hoc Networks? *IEEE Communications Magazine*, pages 130–137, June 2001.
- [87] J. L. Sobrinho, R. de Haan, and J. M. Brazio. Why RTS-CTS is not your Ideal Wireless LAN Multiple Access Protocol. In *Proceedings of IEEE WCNC*, New Orleans, LA, March 2005.
- [88] Giuseppe Bianchi. Performance Analysis of the IEEE 802.11 Distributed Coordination Function. *IEEE JSAC.*, 18(3):535–47, March 2000.
- [89] <http://www.isi.edu/nsnam/ns/>.
- [90] J. Wu and H. Li. On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks. *Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pages 7–14, Aug 1999.
- [91] A. Qayyum, L. Viennot, A. Laouiti. Multipoint Relaying for Flooding Broadcast Messages in MobileWireless Networks. In *Proceedings of IEEE HICSS*, Big Island, HI, January 2002.

- [92] G. Toussaint. The Relative Neighborhood Graph of a Finite Planar Set. *Pattern Recognition*, 12(4):261–8, 1980.
- [93] J. Cartigny, et al. Localized LMST and RNG Based Minimum-energy Broadcast Protocols in Ad hoc Networks. *Ad Hoc Networks*, 3(1):1–16, 2004.
- [94] F. Ingelrest, D. Simplot-Ryl and I. Stojmenovic. Target Transmission Radius over LMST for Energy-Efficient Broadcast Protocol in Ad Hoc Networks. *In Proceedings of IEEE ICC*, Paris, France, June 2004.
- [95] A. Gkelias. *Performance analysis of random access schemes for wireless ad hoc networks*. PhD thesis, Kings College London, Centre for Telecommunications Research, UK, September 2005.
- [96] Silicon Laboratories. C8051F320/1 Datasheet. *Preliminary Rev. 2.1*, 12/2003.

Bibliography

Bibliography

Optimisation des protocoles de routage et d'accès au canal pour allonger de la durée de vie des réseaux sans fil de capteurs

Résumé

L'allongement de la durée de vie d'un réseau de capteurs requiert des optimisations sur deux niveaux complémentaires : local et global. Sur le niveau local, chaque nœud doit optimiser sa consommation d'énergie pour allonger sa durée de vie, et sur le niveau global, les nœuds doivent coopérer ensemble pour optimiser la gestion globale des ressources en énergie.

Dans cette thèse nous nous sommes attaqués à ces deux problèmes par l'optimisation des protocoles de communications. Notre contribution a principalement touché les protocoles MAC et de routage.

Au niveau de la couche MAC, nous avons travaillé sur la réduction de deux principales sources de perte d'énergie : les collisions et l'écoute inutile. Nous avons différencié deux types de collisions : celles causées par les nœuds cachés et celles causées par les nœuds visibles, et deux formes d'écoute inutile : celle résultant de la réception des messages inutiles et celle provenant de la réception des signaux inutiles. Pour toutes ces formes de perte d'énergie, nous avons proposés des solutions pertinentes.

Au niveau de la couche routage, nous avons proposé un protocole permettant de combiner deux techniques d'allongement de durée de vie du réseau : la réduction du surcoût des protocoles et l'équilibrage de charge entre les nœuds pour leur éviter une mort prématurée.

Notre contribution dans cette thèse non seulement fournit des protocoles MAC et des protocoles de routage efficaces en énergie et optimisés les uns indépendamment des autres, mais aussi une ébauche de conception inter couches dans laquelle les protocoles sont optimisés conjointement.

Mots clés : Réseaux de capteurs, Protocoles de communication, MAC, Routage, Économie de consommation d'énergie.

Optimizing Routing and Channel Access Protocols to Extend the Lifetime of Wireless Sensor Networks

Abstract

Extending the lifetime of a sensor network requires optimizations at two complementary scopes: local and global. At the scope front, each node should optimize its own energy consumption to maximize its life span. At the global scope, nodes should cooperate to optimize the global usage of energy resources.

In this thesis, we have addressed both problems through the optimization of communication protocols. Our contribution mainly concerns MAC and Routing protocols.

At the MAC layer, we have tackled the problem of reduction two major sources of energy waste: collisions and overhearing. We have differentiated two types of collisions: those caused by hidden nodes and those caused by visible nodes, and two forms of overhearing: that resulting from the reception of irrelevant messages and that resulting from the reception of useless signals. For all of these identified sources of energy waste, we have proposed solutions that improve the networks lifetime.

At the routing layer, we have proposed a protocol capable of combining two lifetime extension techniques: reducing protocols' overhead and load balancing among nodes to avoid their premature death.

Our contribution in this thesis not only provides more energy-efficient MAC and routing protocols that are optimized independently from each other, but also a cross layer design in which MAC and routing protocols are jointly optimized.

Keywords : Sensor Networks, Communication Protocols, MAC, Routing, Energy Saving.

Discipline : Informatique, Systèmes et Communications

Laboratoire : LSR-IMAG – Équipe Drakkar. BP 72, 38402 Saint Martin d'Hères Cedex, France